Universidad Zaragoza
1542

Trabajo Fin de Grado

Sistema de Ayuda a la Elección de la Alineación de un Partido de Fútbol Profesional

Support System for Choosing a Suitable Lineup for Professional Football Matches

Autor

Óscar Potrony Compaired

Director

Sergio Ilarri Artigas

ESCUELA DE INGENIERÍA Y ARQUITECTURA
2019

Sistema de Ayuda a la Elección de la Alineación de un Partido de Fútbol Profesional

RESUMEN

Cada vez es mayor la competencia entre los equipos de fútbol, y la línea entre vencer o salir derrotado, cada vez más delgada. Es por ello que hay que buscar nuevas alternativas para tratar de sobresalir del resto y optimizar el rendimiento de todo el equipo. Ahí es donde entra la ciencia de datos, que está empezando a ser aplicada a este deporte. En este ámbito se enmarca el presente trabajo, que sigue toda la creación de un proyecto completo de análisis de datos cuya última finalidad es ofrecer la mejor alineación al entrenador de un equipo de fútbol, con el objetivo de maximizar sus probabilidades de ganar el partido.

Tras analizar el estado del arte de la ciencia de datos aplicada al deporte y conocer de primera mano muchas de sus distintas aplicaciones, el primer paso ha sido obtener de repositorios públicos los datos de partidos de las grandes ligas durante varias temporadas, analizando su estructura y calidad.

A continuación, se ha diseñado el almacén de datos siguiendo la metodología de Kimball, formado por varios *data marts* en función de los datos disponibles. Después, se han diseñado los procesos ETL (*Extract*, *Transform and Load*) necesarios para integrar los datos, con la ayuda de la herramienta KNIME. En ellos es donde más tiempo se ha invertido, especialmente en los que se centraban en tratar los datos de los jugadores de todos los conjuntos de datos. En los propios procesos se han introducido los datos en el almacén de datos, implementado utilizando Oracle 10g Express Edition.

El siguiente paso ha sido explotar el almacén de datos de varias formas. Primero, se ha realizado una serie de consultas analíticas para obtener resultados de distintos ámbitos. Después, se han realizado informes de visualización con la herramienta Microsoft PowerBI que sirvan de apoyo a la elección de alineaciones de un entrenador. Finalmente, se han aplicado técnicas de minería de datos para obtener alineaciones, creando un programa orientado al entrenador y evaluando los algoritmos propuestos en función de las alineaciones reales.

En conclusión, se ha entrado en una vía de investigación en el campo del *data science* aplicado al deporte poco transitada iniciando las distintas etapas de un desarrollo completo de análisis de datos, con el deseo de que se profundice más en la misma con ayuda de este proyecto. Por ello, el código y los esquemas desarrollados se han dejado disponibles en un repositorio de GitHub.

Índice

Capítulo 1

Introducción

Este trabajo consiste en un proyecto completo de análisis de datos aplicado al fútbol, desde la obtención de datos y la creación del almacén de datos (para preservación y análisis de información estructurada), hasta la explotación del almacén, tanto mediante informes de visualización como con programas que aplican técnicas de minería de datos. En él se trabaja con datos de partidos de fútbol de las cinco grandes ligas, así como de los eventos acontecidos en ellos, y los equipos y jugadores involucrados, con el objetivo final de facilitar información clave al entrenador de un equipo de fútbol, que entre otras cosas pueda ayudarle a confeccionar una alineación para un partido concreto, con una presentación adecuada de la información. En la Sección 1.1 se van a tratar los motivos que incentivaron la realización del mismo, sus objetivos en la Sección 1.2, la planificación del tiempo y su organización en la Sección 1.3 y la estructura de este documento, en la Sección 1.4.

1.1. Motivación

En unos tiempos donde cada vez es más difícil sobresalir, la tecnología está empezando a cobrar un papel importante en el deporte. Los equipos grandes la necesitan para mantenerse en el más alto nivel sin decaer y los equipos más modestos se empiezan a atrever a confiarle ciertas decisiones con la ilusión de llegar al primer nivel [1]. Esto se hace más evidente en deportes como el béisbol (con el archiconocido ejemplo de *Moneyball* [2], donde un equipo pequeño consiguió maximizar el rendimiento de sus jugadores gracias a un uso pionero de la estadística) o el baloncesto [3], donde es más sencillo "individualizar" el rendimiento de los jugadores de un equipo que en otros casos como el que ahora nos incumbe, el fútbol.

En el "deporte rey" siempre ha habido un rechazo a la introducción de la tecnología,

con muchos detractores alegando que es preferible mantener el juego "como en los viejos tiempos". Poco a poco este miedo está desapareciendo, con la involucración de herramientas como el famoso Videoarbitraje o VAR [4]. No obstante, es indiscutible que el nivel técnico de los futbolistas es cada vez mayor, pudiendo ser denominados atletas, a diferencia de los jugadores de los años 90, sin ir más lejos. Por ello, estos también han de ayudarse de la tecnología para mantenerse en buena forma, evitar lesiones o saber dosificarse.

Por parte de los clubes, que también imponen estos sistemas de prevención de lesiones, midiendo todo tipo de datos de sus jugadores mediante sensores en sujetadores deportivos o chips en las botas, se está empezando a acoger la tecnología para mejorar el rendimiento de sus jugadores o fichar otros en función de las carencias del equipo, el presupuesto disponible y las características de miles de jugadores en el mercado, incluso con equipos de matemáticos analizando los datos [5]. Hay entrenadores de primer nivel que vienen usando estas técnicas desde hace años, como Rafa Benítez, ganador de la *Champions League* con el *Liverpool* [6], y hay referentes en el campo de la dirección deportiva, como Monchi, que están abrazando estas nuevas herramientas [7]. Pero, al igual que ellos, hay trabajadores de equipos pequeños que empezaron hace tiempo y han ido escalando divisiones gracias a su buen hacer y su capacidad para ser visionarios [8].

A raíz de todo esto, han ido surgiendo empresas que obtienen, manejan y venden todo tipo de parámetros de miles de partidos semanales, como Opta [9], otras que permiten el visionado de los partidos con datos superpuestos, como Wyscout [10], y grandes empresas que se han metido de lleno en el mercado, como SAP con su SAP Sports One [11], que tiene un heterogéneo abanico de funciones, desde gestión del equipo de ojeadores hasta gestión del equipo, pasando por la gestión de los propios jugadores, controlando su rendimiento, su estado de forma y su propensión a lesiones. En España también se está tratando este campo: un ejemplo de ello es la *startup* madrileña Driblab [12].

Por todo esto, se ha decidido dar un pequeño paso con este proyecto hacia este mercado en alza, en el que todavía no se han asentado los grandes competidores, creando una herramienta para la gestión de equipos y futbolistas de las grandes ligas, ampliable y con potencial.

1.2. Objetivos

A continuación se exponen los objetivos a alcanzar con la realización del trabajo:

- Diseñar e implementar un almacén de datos con datos de partidos de fútbol de las grandes ligas para preservar información histórica estructurada y poder analizarla.
- Analizar las fuentes de datos e integrarlas en un almacén de datos.
- Explotar el almacén de datos mediante consultas que puedan interesar a un entrenador o a un director deportivo.
- Realizar informes de visualización a partir del almacén de datos para mostrar información clave a un entrenador.
- Aplicar técnicas de minería de datos para sugerir alineaciones a un entrenador.
- En definitiva, se pretende realizar un proyecto de análisis de datos completo.
- Ampliar conocimientos en temas como minería de datos, creación de informes, diseño de procesos ETL, etc.
- Conocer las aplicaciones actuales y vías de investigación futuras de la ciencia de datos en el fútbol.

Como se puede comprobar, en general se pretende ampliar conocimientos sobre análisis de datos y todas las fases de proyectos centrados en el mismo.

1.3. Organización y tiempo dedicado

En esta sección se va a comentar todo lo relacionado con la organización y el tiempo dedicado al proyecto, incluyendo la planificación del mismo.

Para planificar el proyecto en función de las semanas y los apartados principales del mismo, se realizó un Diagrama de Gantt [13] en Microsoft Excel [14], como se puede ver en la Figura 1.1. Para ello, se tuvieron en cuenta una serie de circunstancias, además de las fechas iniciales y finales del proyecto y de la estimación de lo que costaría realizar cada tarea: se pretendía obtener y analizar los datos antes de Navidad pero hasta una vez finalizados los exámenes de enero no se continuaría con el proyecto. Después, y a excepción del principio del segundo semestre, durante el mismo se le dedicaría el tiempo a las asignaturas, retomando el proyecto a mediados

de mayo, cuando se terminara con éstas. Finalmente, se le dedicaría todo el tiempo al proyecto a excepción de unos diez días en septiembre, dedicados a un examen pendiente.

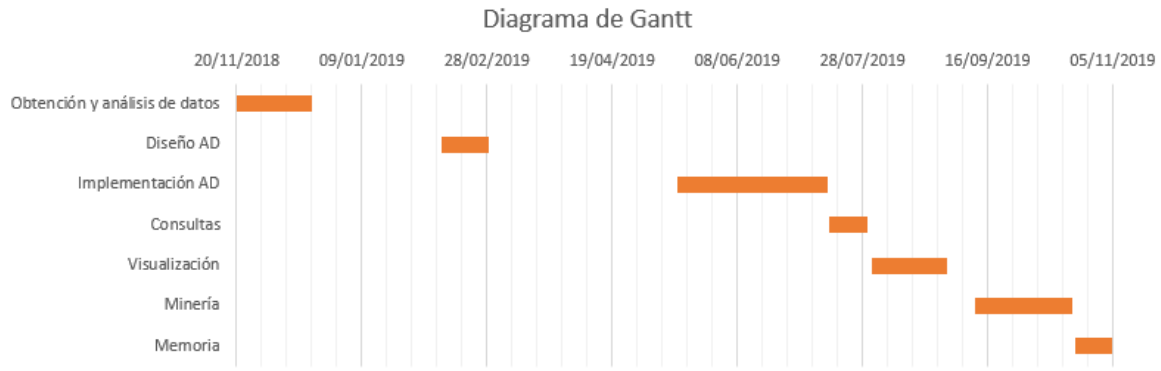

Figura 1.1: Diagrama de Gantt.

En la Tabla 1.1 puede verse un resumen de las horas que finalmente se le dedicaron a cada apartado. El desglose semanal de estas horas se incluye en el Anexo A. Algunos apartados se alargaron más de lo esperado por no estimar bien el tiempo requerido o por circunstancias personales, pero en general se cumplió con la planificación.

Obt	Dis	Imp	Con	Vis	Min	Mem	**Total**
10	18	233	50	60	95	79	**545**

Tabla 1.1: Resumen de tiempo dedicado.

En cuanto a la organización del proyecto, las tareas se gestionaban en un proyecto personal creado en Todoist [15], de forma que ninguna fuera pasada por alto. Los ficheros del proyecto se subían constantemente a la nube (en concreto, a Google Drive [16]), como medida de precaución ante posibles pérdidas de información. Asimismo, los ficheros más importantes (con los que se podría reconstruir todo el proyecto rápidamente), se iban guardando en un *pen drive* (con control de versiones) a buen recaudo, por si ocurría algún problema con Google Drive. Para la gestión de tiempo dedicado y para crear el diagrama de Gantt se utilizó Microsoft Excel [17].

1.4. Estructura de la memoria

En esta sección se va a comentar el contenido de cada capítulo y de cada anexo de este documento.

En este capítulo se ha presentado la motivación del proyecto, así como los objetivos planteados y la planificación y organización del mismo. Después, en el Capítulo 2, se hace un resumen del estado del arte actual de la tecnología aplicada al deporte y, más concretamente, del *Big Data* aplicado al fútbol. También se comentan y explican las tecnologías utilizadas a lo largo de todo el proyecto. En el Capítulo 3 se explica todo lo relacionado con el diseño del almacén de datos, desde la obtención de los datos hasta la creación del modelo. Luego, en el Capítulo 4, se comenta la instalación del sistema gestor de bases de datos y la implementación del propio almacén de datos, así como del diseño de los procesos ETL y la introducción de los datos en el almacén. En el Capítulo 5 se detalla cómo se puede explotar el almacén de datos: mediante consultas, visualización de informes y minería de datos. Por último, en el Capítulo 6 se exponen las conclusiones personales y del proyecto y el trabajo futuro que ofrece el mismo.

En cuanto a los anexos, en el Anexo A se muestra la división de horas dedicadas a cada apartado durante cada semana desde el inicio del proyecto hasta su entrega. En el Anexo B se detalla el análisis preliminar de los conjuntos de datos elegidos, así como los esquemas relacionales de los mismos y sus limitaciones, entre las que se enumeran también los datos que no se han conseguido y se desearía tener para poder realizar análisis más ricos. En el Anexo C se muestra la matriz de bus del almacén y se explican los identificadores especiales de las dimensiones y los posibles valores de la tabla de dimensión DIM_TipoEvento. A continuación, en el Anexo D se incluyen las consultas de explotación más relevantes creadas. En el Anexo E aparece todo lo relacionado con los cuadros de mandos de los informes de visualización: explicación, creación y muestra. Finalmente, en el Anexo F se incluyen detalles referentes a las tareas de minería de datos realizadas: una traza del programa de cara al usuario, ejemplos de todas las alineaciones para un partido concreto, evaluaciones de los algoritmos propuestos y la creación del modelo de regresión logística.

Capítulo 2

Contexto Tecnológico

En este capítulo se enmarca lo relacionado al estado del arte y las tecnologías utilizadas en este proyecto. En la Sección 2.1 se va a exponer el resultado de la investigación realizada sobre el estado del arte de este tipo de proyectos y su temática, y en la Sección 2.2 se van a detallar las tecnologías que se han utilizado para el proyecto.

2.1. Estado del arte

Actualmente existen muchas vías abiertas para aplicar ciencia de datos en el fútbol. Una de las razones es la gran cantidad de datos que se generan, ya que participan muchos jugadores durante 90 minutos, además de los entrenamientos. A continuación se van a presentar algunas de estas vías.

Una de las investigaciones más interesantes que se han estudiado trata sobre el análisis de la táctica del juego, utilizando todo tipo de parámetros [18]: tratan de ir más allá de las estadísticas sencillas que se han utilizado siempre y ya parecen insuficientes (como las variables de pase o posesión), empezando a incluir parámetros contextuales, fisiológicos, históricos, técnicos, etc. En el artículo citado se menciona el análisis de la táctica en función del centroide del equipo sobre el campo, de la fisiología de los jugadores (para la que no se ha encontrado una conexión clara) o incluso creando redes neuronales que representen los pases entre los jugadores. Sin embargo, estos análisis no tienen una conexión conceptual, por lo que no hay un modelo que las aúne. Para su autor, el futuro en este campo es estimulante, ya que anteriormente el problema principal era la falta de datos (y la falta de confianza en los pocos que se tenían, que no llegaba a los estándares mínimos de calidad), ya inexistente.

Una vía mucho más transitada que la anterior es la predicción de resultados. En algunos trabajos, [19], ésta se estudia utilizando múltiples técnicas de minería de

datos. Otro trabajo trata de encontrar los factores clave que lleven a la conclusión de que se va a ganar un partido, para saber qué variables son las más importantes a tratar para estas predicciones y la influencia de las mismas [20].

En relación al área anterior, otros tratan de mejorar el rendimiento global de los equipos, no centrándose únicamente en el resultado final de los partidos. Otro trabajo estudiado se centra en indicadores de rendimiento centrados en los pases del equipo. Los autores consiguieron obtener clasificaciones en simulaciones de varias ligas europeas muy similares a las reales, utilizando únicamente métricas relacionadas con los pases y creando un indicador propio llamado 'H' que las aunaba [21].

También hay trabajos que tratan sobre obtener la mejor alineación posible en un partido de fútbol [22]. Éste en concreto se divide en dos pasos: primero calculan tasas sobre cómo de beneficioso es cada jugador para su equipo en función de sus características y sus victorias, mediante una red neuronal semi supervisada. Después, mediante otra red neuronal con los onces de los dos equipos (el seleccionado y el oponente, para el que se supone un once conocido, y ambos con formación "4-4-2"), calculan la probabilidad de victoria.

Además, hay muchas otras subáreas que se están investigando actualmente, como la prevención de lesiones, el rendimiento de jugadores, la asignación de valor a jóvenes promesas, etc. Incluso se han realizado estudios sobre el efecto de las decisiones arbitrales en el resultado de los partidos [23], intentando determinar el posible sesgo introducido al sacar tarjetas amarillas o rojas.

2.2. Tecnologías utilizadas

En esta sección se van a enumerar y explicar todas las tecnologías utilizadas en el proyecto, así como algunas alternativas que se planteó utilizar. También se enumeran las metodologías seguidas. Cabe destacar que todo el proyecto se ha llevado a cabo en el ordenador personal del autor, cuyo sistema operativo es Windows 10 Home Edition.

Las tecnologías principales utilizadas para el proyecto son las siguientes:

- Sublime Text 3 [24]: es el editor de texto por defecto utilizado en todo el proyecto. Sublime tiene múltiples opciones para aumentar la comodidad y productividad del usuario, entre las que se encuentra la sintaxis de múltiples lenguajes de programación (como SQL o R), ya sea de forma nativa o mediante paquetes externos.

- Overleaf [25]: es el editor online de textos LaTeXdonde se ha creado la memoria del proyecto. La almacena en la nube, es colaborativo, permite control de versiones, compilar en cualquier momento para observar el resultado, y tiene control de gramática y conteo de palabras.

- Kaggle [26]: es una comunidad *online* donde se pueden encontrar conjuntos de datos de todos los ámbitos subidos por los usuarios, para que otros los usen y compartan sus soluciones, para discutir sobre ellas o utilizarlas.

- DBDAp [27]: es una herramienta de modelado de esquemas conceptuales de bases de datos y almacenes de datos, creada en la Universidad de Zaragoza con apoyo de proyectos de fin de estudios.

- Oracle 10g Express Edition [28]: es un sistema gestor de bases de datos gratuito que posee ciertas limitaciones frente a versiones comerciales de Oracle, relacionadas con el tamaño de los datos a almacenar y el número de usuarios que permite, lo que no son problemas para este proyecto en particular.

- Docker [29]/Docker Hub [30]: Docker es una tecnología de contenedores, cuya idea es crearlos ligeros y portables para que las aplicaciones software contenidas puedan ejecutarse en cualquier máquina con Docker, independientemente de otros requerimientos. Docker Hub, por su parte, es el repositorio de contenedores oficial de Docker, donde los usuarios (o las empresas) pueden subir imágenes de sus propios proyectos, para que otros puedan utilizarlos.

- Virtualbox [31]: es una herramienta de virtualización abierta que permite la ejecución de múltiples máquinas virtuales, con diferentes sistemas operativos.

- KNIME [32]: es una herramienta de diseño de procesos ETL de forma visual, de forma que mediante distintos nodos se pueda limpiar y dar formato a los datos, pudiendo ver la salida de cada nodo en cualquier momento.

- IntelliJ IDEA [33]: es un entorno de programación para Java muy completo, con características como múltiples lenguajes, autocompletado o control de versiones.

- Microsoft PowerBI [34]: es una herramienta de *Business Intelligence*, en la que se pueden crear informes orientados a humanos a partir de los datos que se le dan como entrada, pudiendo relacionarlos y manejarlos gracias a PowerQuery. Los informes se componen de cuadros de mandos, en los que se pueden incluir gráficos, tablas, etc. Los gráficos pueden obtenerse, además de los que aparecen de forma nativa, de su *marketplace* propio.

- RStudio [35]: es el entorno de programación del lenguaje R utilizado para la minería de datos y los análisis preliminares de los conjuntos de datos. Permite múltiples ventanas, con las variables, las gráficas, los documentos y la consola en un simple vistazo.

Las alternativas que se plantearon son:

- Tableau [36]: es una herramienta de *Business Intelligence*, que no se ha utilizado debido a que ya se tenía cierta experiencia con PowerBI y se conocían más sus capacidades. Es de pago, aunque cuenta con versión de prueba gratuita.

- R [37]: la terminal de R es más simple que RStudio y tiene muchas menos funcionalidades, por lo que se decidió no optar por ella.

- Weka [38]: es una herramienta de minería de datos y aprendizaje automático con herramientas de preparación de datos, clasificación, reglas de asociación, etc. Finalmente se utilizó solo R por su mayor polivalencia.

Algunas tecnologías utilizadas de forma casual son Wikidata [39], SQLiteStudio [40] o DAX Studio [41]. Otras tecnologías, más habituales en cualquier proyecto, son Microsoft Excel [17], Todoist [15] o Google Drive [16].

Para realizar el proyecto se ha utilizado la metodología CRISP-DM (*Cross Industry Standard Process for Data Mining*) [42], un modelo estándar abierto que proporciona una descripción normalizada del ciclo de vida de un proyecto estándar de *data science*. Este ciclo de vida cuenta con seis fases principales, dependientes entre sí: comprensión del negocio, comprensión de los datos, preparación de los datos, modelado, evaluación e implantación.

Para la creación del almacén de datos se ha seguido la metodología de Kimball [43], como se explica detalladamente en la Sección 3.2.1. La alternativa habría sido la metodología de Inmon [44], descartada porque se adapta peor al problema planteado en este proyecto y porque la de Kimball ya se había utilizado en la asignatura "Almacenes y minería de datos".

Capítulo 3

Diseño del almacén de datos

En este capítulo se van a detallar, en la Sección 3.1 la fase de obtención de datos, y en la Sección 3.2 la del diseño del almacén de datos, donde se explica la metodología utilizada y se detallan las tablas que lo conforman.

3.1. Obtención de datos

El primer paso previo a comenzar a diseñar el almacén de datos ha sido buscar conjuntos de datos públicos que pudieran resultar de interés. Para ello, primero se realizó una búsqueda y clasificación preliminar de conjuntos de datos para conocer las posibilidades que podrían ofrecer los existentes, sin entrar en mucho detalle. Se descubrió que había una cantidad nada desdeñable de conjuntos, tanto de partidos de distintas ligas como de jugadores y sus características. Se clasificaron en diferentes apartados según de qué trataran, y se indicó qué características poseía cada uno, como los años y las ligas que abarcaban sus datos, si tenía datos del partido y/o de los jugadores, etc. A partir de esta clasificación preliminar, se realizaron dos clasificaciones más detalladas:

En la primera se tuvieron en cuenta tanto aspectos de los equipos (información básica, sus jugadores, etc.) como de los partidos, con datos generales (como el resultado), estadísticas generales (como el número de saques de esquina), alineaciones, goleadores o incluso eventos más concretos, como una falta cometida por un jugador sobre otro.

En la segunda se incluyeron los conjuntos de datos relacionados con jugadores, tanto los que contenían estadísticas reales como los que tenían datos procedentes de la saga de videojuegos FIFA. Ésta contenía apartados como datos personales, características (valores numéricos para distintas habilidades), puntuación global, valor de mercado, etc.

Simultáneamente, se ideó una serie de factores que influyen a la hora de elegir una

alineación y que se podrían manejar con los datos que se observaron en la búsqueda preliminar. Para cada factor se marcaron los datos necesarios, para simplificar la decisión de qué datos eran más necesarios gracias a esta ayuda visual.

Finalmente se decidió utilizar dos conjuntos de datos e integrarlos. Ambos se encontraron en Kaggle [26]. Al integrarlos, se acabaron manteniendo los datos de las cinco ligas más importantes (las primeras divisiones de España, Inglaterra, Italia, Alemania y Francia) durante las temporadas de las que había información en ambos conjuntos de datos (desde la temporada 2011/2012 hasta la temporada 2015/2016, ambas incluidas).

El primer conjunto de datos escogido se llama "European Soccer Database" [45] y su autor es Hugo Mathien. Éste contiene información de equipos, partidos y jugadores (esta última extraída del videojuego FIFA). Venía en formato base de datos SQLite, por lo que para estudiarlo inicialmente se utilizó el programa SQLiteStudio [40], pero posteriormente se exportó manualmente a formato CSV (Figura 3.1).

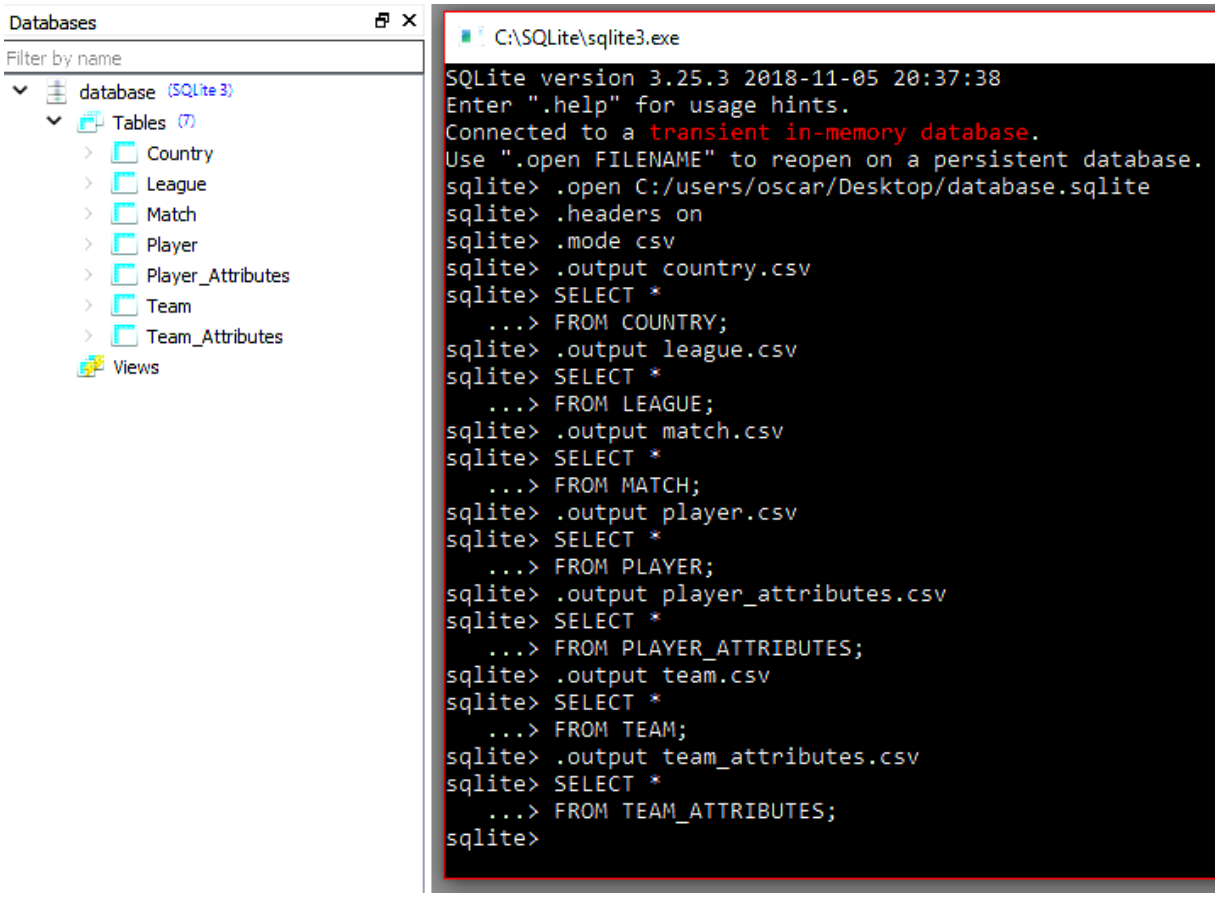

Figura 3.1: Exportación de datos de SQLite a CSV.

El segundo, de Alin Secareanu, se llama "Football Events" [46]. Contiene eventos de partidos de las grandes ligas, con atributos como las personas involucradas en él.

A simple vista se observó que no estaba muy estandarizado (ya que en cada evento mencionaba a los jugadores con texto plano, por ejemplo), por lo que habría que tratarlo en profundidad en los procesos ETL.

Además, se optó por recuperar los estadios de Wikidata, para utilizar otro recurso diferente y porque proporcionan datos de interés para análisis. En la consulta se obtuvieron los nombres de todos los estadios de fútbol existentes en Wikidata, junto a su club, sus coordenadas, su capacidad y su país (Figura 3.2).

```
SELECT ?clubLabel ?venueLabel ?capacidad
    ?ligaLabel ?paisLabel ?coordinates WHERE {
  ?club wdt:P31 wd:Q476028.
  ?club wdt:P115 ?venue.
  ?club wdt:P118 ?liga.
  ?club wdt:P17 ?pais.
  ?venue wdt:P625 ?coordinates.
  ?venue wdt:P1083 ?capacidad.
  SERVICE wikibase:label
        { bd:serviceParam wikibase:language ''en''. }
}
```

Figura 3.2: Consulta para obtener información de los estadios de fútbol de Wikidata.

Después, se realizó un análisis preliminar de todos los datos. En general eran bastante correctos. Sin embargo, hay una serie de limitaciones, como que faltan algunos datos de ciertos partidos o que algunos son erróneos. Se puede leer el estudio de forma más detallada en el Anexo B.

3.2. Diseño del almacén de datos

En esta sección se van a tratar los pasos que se siguieron para diseñar el almacén de datos y se van a explicar todas las dimensiones y tablas de hechos, así como decisiones de diseño tomadas en las mismas. Cabe destacar que para el diseño del esquema en estrella se ha utilizado el programa DBDAp [27]. Además, en el Anexo B se han detallado las limitaciones de los conjuntos de datos iniciales y los datos de los que se habría deseado disponer y no ha sido posible. Por estas limitaciones, ha sido necesario crear tuplas especiales en varias dimensiones para indicar que no se tiene la información. Estas tuplas están anotadas y explicadas en el Anexo C.

3.2.1. Metodología de Kimball

Se decidió seguir la metodología de Kimball [43] para realizar el diseño del almacén de datos (en detrimento de alternativas como la metodología de Inmon [44]), de forma

que el almacén es una integración de dos *data marts* claramente diferenciados: el de eventos y el de alineaciones. Posteriormente se tratarán los demás, donde las tablas de hechos son agregadas.

1. Selección de los procesos de negocio

En el *data mart* de eventos, la actividad clave es tener un registro de todos los eventos acontecidos en cada partido.

Sin embargo, en el *data mart* de alineaciones la necesidad más importante es la de saber las alineaciones que cada equipo ha sacado en cada partido.

2. Declaración del grano

Como en el *data mart* de eventos se pretendía guardar todos los eventos, cada tupla ha de ser un evento, por lo que el grano es cada evento de cada partido.

Por otro lado, el grano del *data mart* de alineaciones es mayor, siendo el grano cada partido.

3. Elección de las dimensiones

A partir de las declaraciones de los granos anteriores, se van a definir todas las dimensiones a utilizar.

En el *data mart* de eventos aparecen todas las dimensiones que se van a utilizar posteriormente. En este caso, las dimensiones primarias (determinadas por el grano) son: DIM_Fecha, DIM_Equipo, DIM_Jugador, DIM_TipoEvento, DIM_DetallesEvento y la dimensión degenerada minuto. Las secundarias (que toman un único valor por cada combinación de las primarias) son: DIM_Liga, DIM_Estadio, DIM_VersionEquipo, DIM_VersionJugador, DIM_DatosJugador y las dimensiones degeneradas temporada y jornada.

En el *data mart* de alineaciones, en cambio, solo DIM_Fecha y DIM_Equipo son dimensiones primarias, mientras que DIM_Liga, DIM_Estadio, DIM_VersionEquipo, DIM_Jugador, DIM_VersionJugador, DIM_DatosJugador y las dimensiones degeneradas temporada, jornada, formacionCasa y formacionFuera son las secundarias.

En la Sección 3.2.2 se desarrollan más los propósitos y características de cada una de las dimensiones. Además, en el Anexo C se ha incluido la matriz de bus del almacén de datos, de forma que se pueda comprobar de una forma mucho más visual qué dimensiones incluye cada *data mart*.

4. Identificación de los hechos

Fact_EventosConocidos, la tabla de hechos del *data mart* de eventos, es una tabla de hechos sin métricas, ya que la información necesaria viene dada por los valores de dimensiones que registra.

Por su parte, los hechos de Fact_AlineacionesConocidas, la tabla de hechos del *data mart* de alineaciones, son seis arrays de catorce elementos cada uno como máximo (un elemento por cada jugador de cada equipo, incluyendo los once jugadores titulares y los tres suplentes). Estos arrays son minutosJugadosCasa, minutosJugadosFuera, coordenadaXCasa, coordenadaYCasa, coordenadaXFuera y coordenadaYFuera. Todos ellos son aditivos.

Los elementos de las tablas de hechos, así como las decisiones relacionadas con ellos, son tratados en la Sección 3.2.3.

3.2.2. Dimensiones

En este apartado se van a detallar todas las dimensiones (excepto las dimensiones degeneradas, cuyas explicaciones vienen implícitas en sus nombres), justificando decisiones de diseño de sus atributos e informando de sus características.

Algunas características inherentes a todo el almacén de datos son el uso de claves artificiales (para evitar embeber inteligencia a las propias claves y aislarlas de posibles cambios futuros) o el rechazo a utilizar nulos, empleando en su lugar tuplas especiales con enteros aislados y utilizados únicamente con esa finalidad.

DIM_Fecha

Además de los habituales atributos de día, semana y mes, se ha decidido guardar el día de la semana porque podría resultar interesante para el usuario diferenciar entre jornadas jugadas en fin de semana y las jugadas a mitad de semana. También se ha guardado una cadena de caracteres con el texto de la fecha representado con el formato "dd/mm/aaaa". Se ha decidido relacionar únicamente las fechas de los partidos y los eventos a esta dimensión, en detrimento de las fechas de recopilación y de nacimiento de otras dimensiones. Esta decisión se sustenta en que se ha preferido que las fechas de DIM_Fecha (Figura 3.3) sean exclusivamente fechas de partidos de fútbol.

DIM_Liga

En esta dimensión (Figura 3.4) no hay nada reseñable, tan solo se han guardado el nombre de la liga y el país al que pertenece.

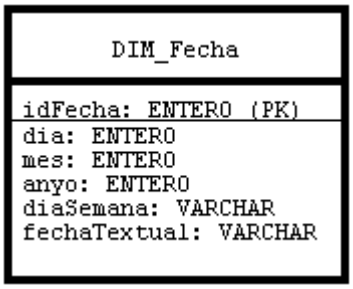

Figura 3.3: Dimensión DIM_Fecha.

Figura 3.4: Dimensión DIM_Liga.

DIM_Estadio

En esta dimensión (Figura 3.5) se ha guardado el nombre, la capacidad y las coordenadas de los estadios, que podrían resultar de interés a los usuarios para dividir partidos por zonas diferentes a países. Todos ellos se han obtenido, como se ha mencionado previamente, de Wikidata. Se han desligado los estadios de los equipos manteniéndolos sin relación e indicando en cada partido y en cada evento el estadio donde ha ocurrido. Se ha creído que esto es más conveniente ya que hay estadios en los que juegan varios equipos como locales, como el Giuseppe Meazza en Milán (donde juegan el AC Milan y el Internazionale), y porque los equipos cambian de estadio. Además, de esta forma es más sencillo tratar con posibles excepciones, como que un equipo juegue como local en el estadio de otro por alguna circunstancia especial [47].

Figura 3.5: Dimensión DIM_Estadio.

DIM_TipoEvento

Esta dimensión (Figura 3.6) inicialmente no iba a existir. Sus atributos eran las métricas en Fact_EventosConocidos, que funcionarían de la misma forma que en la actualidad: si un atributo vale 1, ese atributo ha ocurrido en el evento que se esté consultando (si vale 0, no). A la hora de agrupar funcionaba bien, pero se decidió apartar en una dimensión porque se consideró que podría causar confusión al usuario y dar lugar a errores. Además, de esta forma se ahorra espacio.

Para identificar qué significa cada uno de los atributos de esta dimensión, se puede comprobar el Anexo C, donde se ha explicado el significado de todos los atributos y sus valores.

Figura 3.6: Dimensión DIM_TipoEvento.

DIM_DetallesEvento

Esta dimensión (Figura 3.7) contiene atributos menos importantes sobre los eventos, tales como la localización donde ha ocurrido, o el lugar desde donde se ha realizado un disparo. Se han aprovechado del conjunto de datos original, dándoles otro formato más adecuado para este *data mart*. Se ha incluido un atributo con el tipo de evento acortado para que los usuarios más acostumbrados a tratar con el almacén puedan utilizarlo como atajo para ser más productivos (en las consultas, por ejemplo, en vez de *tipoEventoTextual = 'Segunda Tarjeta Amarilla'*, podrían utilizar *tipoEventoTextualCorto = 'STA'*).

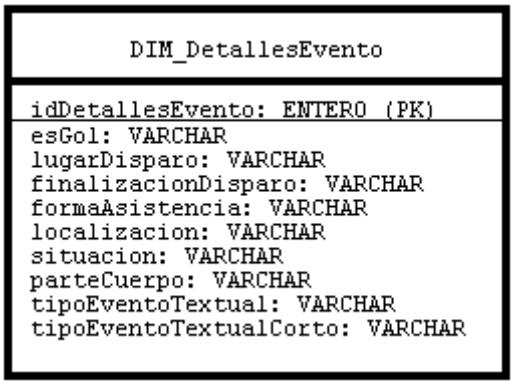

Figura 3.7: Dimensión DIM_DetallesEvento.

DIM_Equipo

Aunque inicialmente residía más información en esta dimensión (que finalmente se dividió en dos, como se explicará en DIM_VersionEquipo), finalmente únicamente guarda el nombre completo y el corto del equipo, así como la clave ajena de la versión actual del propio equipo. Esta es, la última versión del equipo obtenida según las fechas de recopilación. Se puede ver en la Figura 3.8.

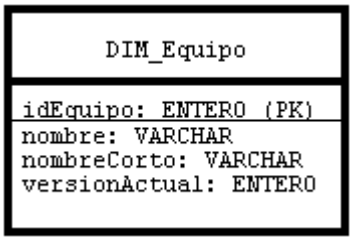

Figura 3.8: Dimensión DIM_Equipo.

DIM_VersionEquipo

Se decidió extraer esta información de DIM_Equipo para que ésta, que presumiblemente iba a ser mucho más consultada por el usuario, fuera más ligera y legible, ya que inicialmente cada tupla de DIM_Equipo era una versión de un equipo, y no un equipo como tal. La forma actual es más entendible. De esta forma, DIM_VersionEquipo (Figura 3.9), que contiene una tupla por cada versión de cada equipo, almacena el id del equipo, la fecha de recopilación de los datos y los propios atributos. Al tener almacenado dicho id, es posible analizar la evolución de un equipo con el paso del tiempo. Los datos de los equipos y de los jugadores han sido obtenidos de varias entregas de la saga de videojuegos FIFA. En estos videojuegos las estadísticas de los equipos y jugadores son variantes en función del rendimiento de estos en la realidad. Por ello, en cada entrega se realizan varios escaneos de todos los equipos y jugadores,

y de cada uno de ellos surge lo que aquí se llama una 'versión'. La fecha en la que se realiza cada escaneo son las llamadas fechas de recopilación.

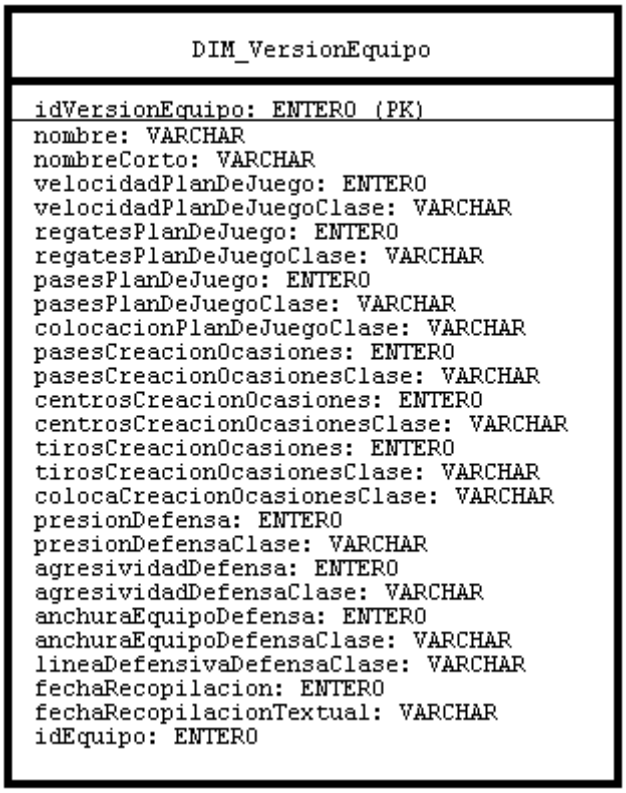

Figura 3.9: Dimensión DIM_VersionEquipo.

DIM_Jugador

Al igual que DIM_Equipo, DIM_Jugador (Figura 3.10) inicialmente almacenaba toda la información de versiones y fue dividida en varias dimensiones para aligerar los procesos de consulta. Finalmente solo cuenta con el nombre del jugador y dos claves ajenas: la de la versión actual del jugador y la de sus datos personales.

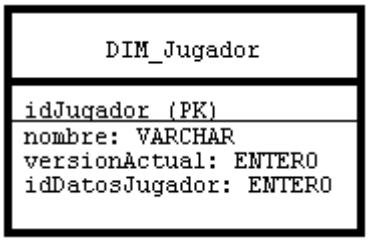

Figura 3.10: Dimensión DIM_Jugador.

DIM_VersionJugador

Con esta dimensión (Figura 3.11) ocurre lo mismo que con DIM_VersionEquipo: cada tupla corresponde a una versión de un jugador y contiene el id del jugador del

que trata la tupla (para poder analizar su evolución), la fecha de recopilación de los atributos y los propios atributos.

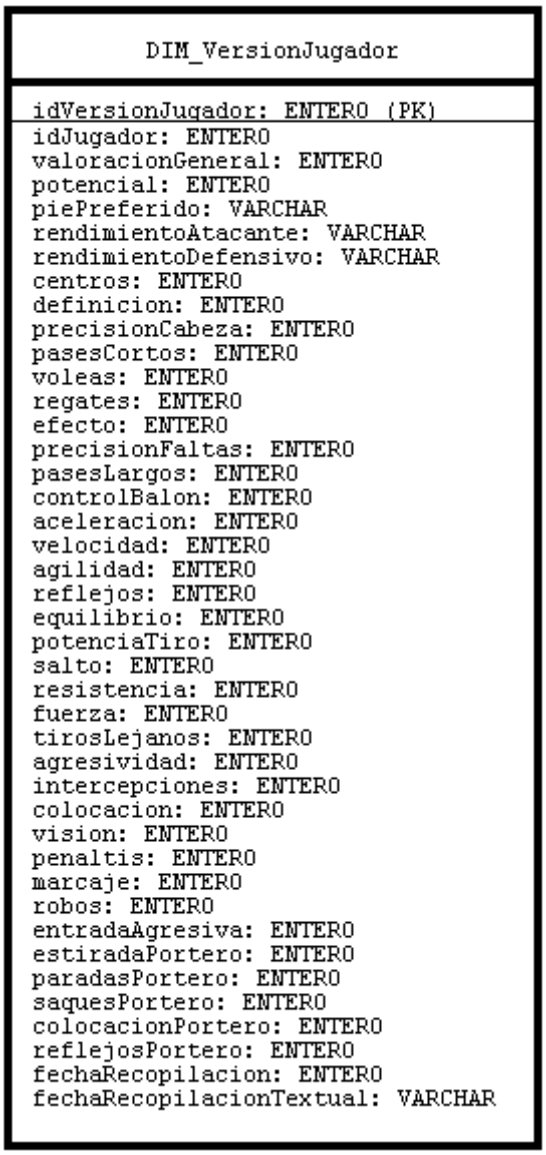

Figura 3.11: Dimensión DIM_VersionJugador.

DIM_DatosJugador

DIM_DatosJugador (Figura 3.12) es una minidimensión [48] extraída de DIM_Jugador, ya que se ha considerado que sus atributos (fecha de nacimiento, peso y altura) van a ser menos consultados por los usuarios, de forma que extraerlos aligera las consultas, además de que algunos de estos atributos pueden cambiar frecuentemente. Se ha decidido incluir las unidades de medida del peso y la altura en el propio nombre del atributo para no dar lugar a confusiones.

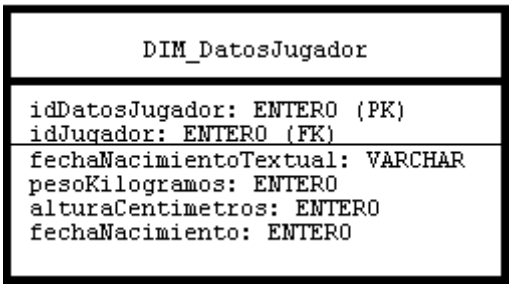

Figura 3.12: Dimensión DIM_DatosJugador.

3.2.3. Tablas de hechos

En este apartado se van a desarrollar las dos tablas de hechos simples, mencionando sus características, detallando sus claves primarias y justificando algunas decisiones de diseño. Dado que hay partidos para los que no se tienen eventos y otros para los que no se tiene información de alineaciones debido a limitaciones de los conjunto de datos originales, se ha añadido el sufijo "Conocidos/as" en los nombres de las tablas de hechos para que los usuarios sean conscientes de esta limitación.

Fact_AlineacionesConocidas

Esta tabla de hechos (Figura 3.13) almacena, en esencia, la información básica del partido que está tratando y la información de los jugadores que lo han jugado. De esta forma, almacena las claves ajenas de los dos equipos, de las versiones de ambos equipos en el momento que se jugó, de los veintiocho jugadores que lo han jugado (catorce de cada equipo a lo sumo), de las versiones de todos ellos en ese momento, de fecha, de liga y de estadio y las dimensiones degeneradas temporada, jornada y las formaciones de cada equipo (una cadena que representa el número de jugadores en cada línea del campo, a excepción del portero). La clave primaria está formada por idEquipoCasa e idFecha. No es necesario nada más para asegurar que es único, ya que un equipo solo puede jugar un partido al día como máximo. Nótese que el tipo "A-14" de la figura indica un array de 14 enteros (con los ids de los once jugadores titulares y los de tres posibles suplentes, como máximo).

En cuanto a las métricas, almacena arrays con los minutos jugados y las coordenadas X e Y de los veintiocho jugadores. Se han guardado arrays de catorce elementos y no catorce atributos para hacerlo más legible y tratar de facilitar el trabajo a los usuarios.

Se manejó la posibilidad de convertir las coordenadas (que indican la posición exacta de cada jugador en el campo) a posiciones concretas (como, por ejemplo, lateral izquierdo o delantero centro). Se ha optado por mantener las coordenadas para que sea más fácil de analizar y menos interpretable, ya que un mismo punto podría considerarse

una posición u otra en función del resto del equipo (en una defensa de cinco el lateral toma una posición similar a la de un interior en una formación más habitual, como el 4-3-3).

No se almacena qué jugador ha sido sustituido por otro porque los cambios del conjunto de datos de eventos no eran del todo fiables (como se comenta en el Anexo B.3) y porque no ha resultado de gran interés.

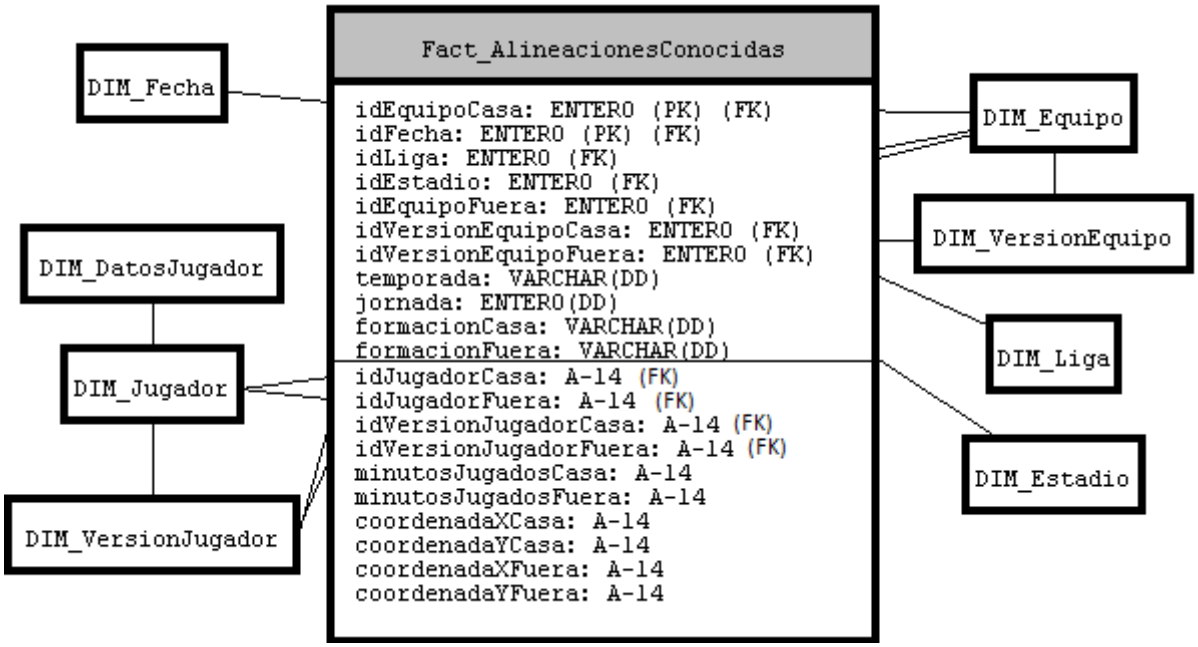

Figura 3.13: Tabla de hechos Fact_AlineacionesConocidas.

Fact_EventosConocidos

Inicialmente, esta tabla de hechos (Figura 3.14) contenía métricas tales como golMarcado, faltaCometida, etc. Estas métricas eran indicadores que advertían de lo que había ocurrido en cada evento. Finalmente, se decidió crear una nueva dimensión con ellos. Así, Fact_EventosConocidos es una tabla de hechos sin métricas [49], donde la información viene dada por los valores de dimensiones que registra. Estos guardan información básica del partido e información acerca del propio evento que se está representando, ya que cada tupla es un evento concreto en un partido concreto. Por ello, la clave primaria está compuesta de varios atributos, ya que se tiene que representar qué partido es (idEquipoCasa + idFecha) y el evento (idJugador1 + idJugador2 + minuto + idTipoEvento + idDetallesEvento), para asegurar que las tuplas tienen claves primarias únicas.

Para indicar qué equipo es el que ha protagonizado el evento, se ha almacenado idEquipoEvento, una clave ajena de DIM_Equipo. Otra posibilidad que se manejó fue almacenar un indicador que señalara cuál de los dos equipos del partido era el

protagonista del evento. Se decidió el método actual porque el otro no mejora el rendimiento total y es menos interpretable por el usuario.

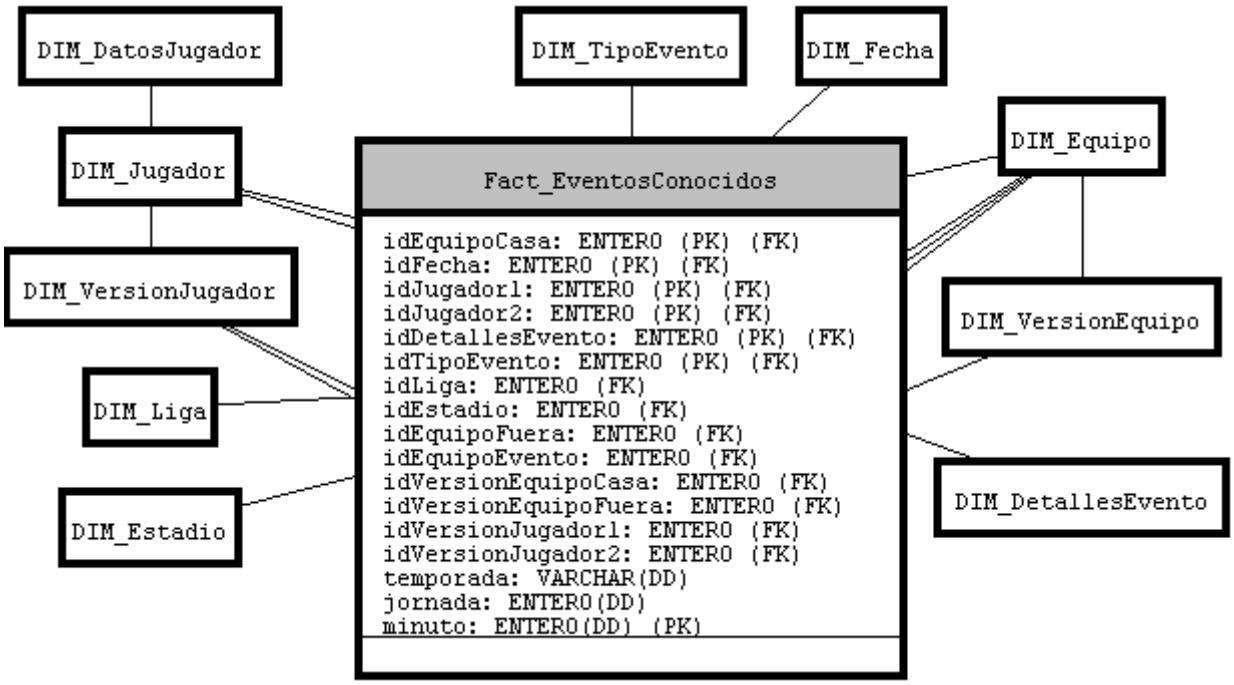

Figura 3.14: Tabla de hechos Fact_EventosConocidos.

3.2.4. Tablas de hechos agregadas

En este apartado se van a detallar los atributos de las tablas de hechos agregadas, creadas en procesos ETL a partir de las dos tablas de hechos ya mostradas. Estas tablas tienen diferentes granos, que se especificarán para justificar sus claves primarias y demás atributos.

AggFact_JugadorPartido

El grano de esta tabla de hechos agregada (Figura 3.15) es cada jugador de cada partido, por lo que contiene información del partido tratado, del jugador tratado, y las métricas que muestran la información de dicho jugador en dicho partido (minutos jugados, coordenadas que indiquen su posición en el campo, goles marcados, faltas cometidas, etc.). Para indicar de qué equipo es el jugador se ha utilizado idEquipoJugador, de una forma semejante a idEquipoEvento en Fact_EventosConocidos. Esta tabla es útil para poder seguir el rendimiento de los jugadores.

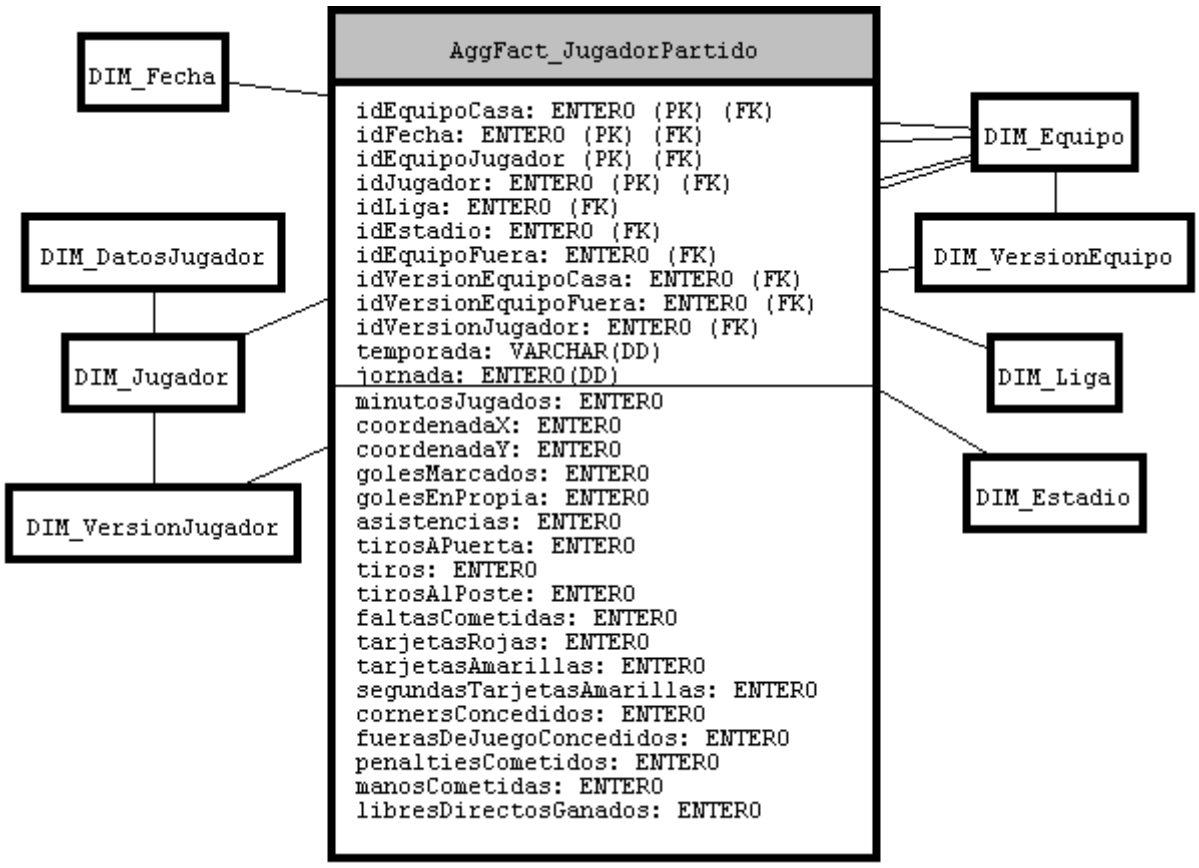

Figura 3.15: Tabla de hechos agregada AggFact_JugadorPartido.

AggFact_Partido

El grano de AggFact_Partido (Figura 3.16) es cada partido, al igual que Fact_AlineacionesConocidas. Por ello, almacena información básica del mismo, información de ambos equipos y, como métricas, las estadísticas de ambos, con el sufijo 'Casa' o 'Fuera' en función del equipo al que se refiera la métrica. Sirve para poder tener toda la información de cada partido sin realizar ninguna operación.

AggFact_EquipoTemporada

En este caso el grano es cada equipo cada temporada. Esta tabla de hechos (Figura 3.17) es la más diferente en cuanto a información almacenada, ya que, además de guardar la información básica del equipo, guarda las estadísticas totales de la temporada del mismo, un array con los ids de todos los jugadores que han jugado para el equipo en la temporada, y el número de partidos ganados, empatados y perdidos. Nótese que el tipo "A-40" de la figura indica un array de 40 enteros (el valor máximo de una plantilla, según los datos, es de 38 jugadores).

Se ha decidido almacenar numPartidosConInfo por la limitación de no tener la información de todos los partidos, como se comenta en el Anexo B.3. También se

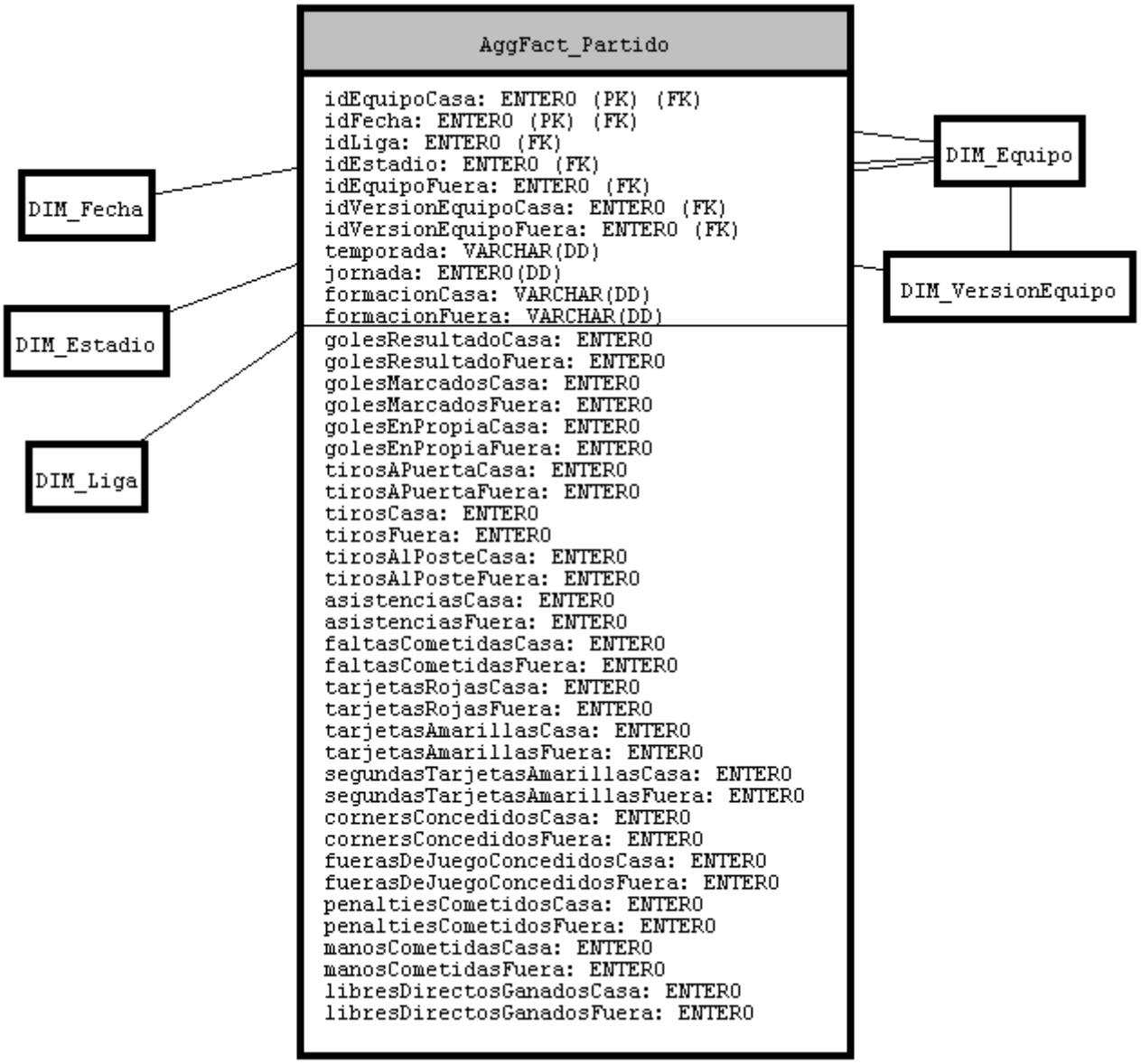

Figura 3.16: Tabla de hechos agregada AggFact_Partido.

guardó formacionMasUtilizada, calculada mediante las formaciones de los partidos, para que el usuario tenga dicha información fácilmente accesible.

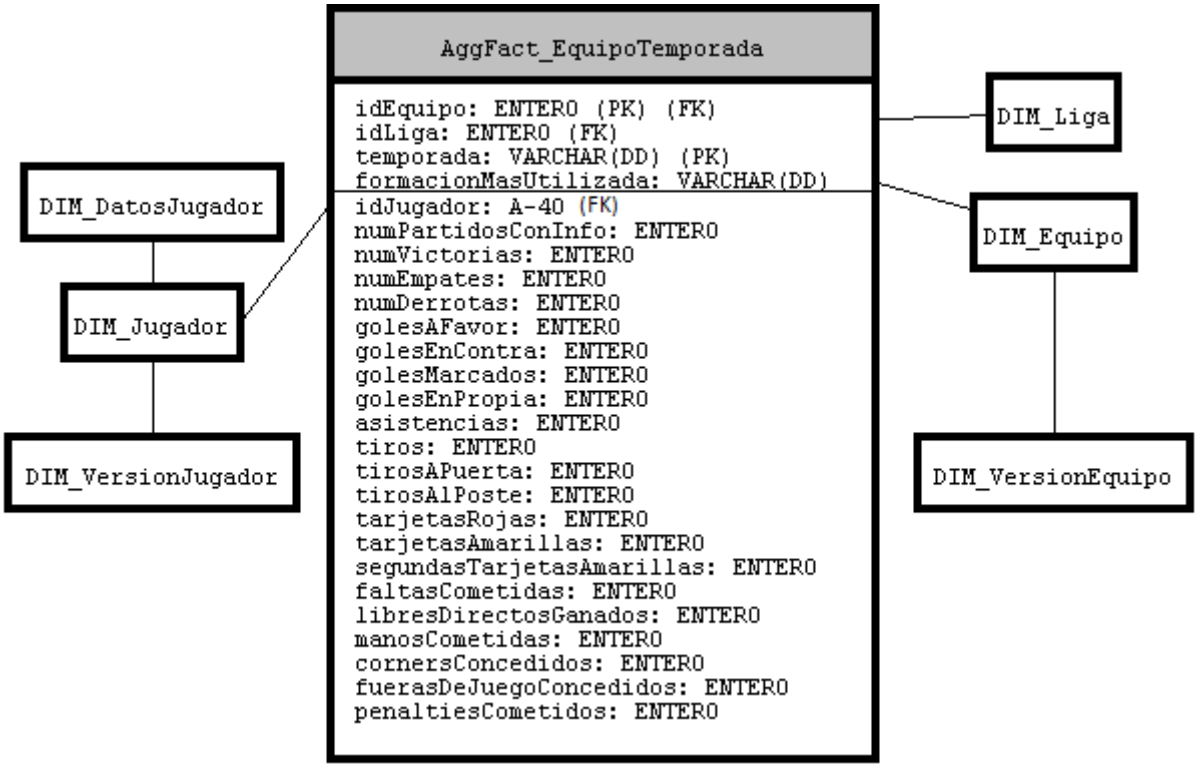

Figura 3.17: Tabla de hechos agregada AggFact_EquipoTemporada.

Capítulo 4

Implementación del almacén de datos

En este capítulo se van a tratar varios aspectos de la implementación del almacén de datos, haciendo especial hincapié en los procesos ETL. En la Sección 4.1 se comenta todo lo relacionado con el sistema gestor de bases de datos, mientras que en la Sección 4.2 se comentan los casos más complicados de solucionar de los procesos ETL, así como una explicación general de los mismos.

4.1. Utilización del sistema gestor de bases de datos

En cuanto a la implementación del propio almacén de datos, el sistema gestor de bases de datos empleado ha sido Oracle 10g Express Edition [28]. Su instalación se realizó utilizando Docker [29] en Windows. Se ha optado por el uso de Docker Toolbox con la virtualización en VirtualBox [31]. El contenedor de Oracle para Docker se obtuvo de Docker Hub [30]. En su momento se obtuvo de un repositorio [50] que actualmente no almacena nada. El autor del contenedor era "dkfi". Actualmente se puede encontrar un contenedor parecido en otro repositorio [51] (de hecho, puede observarse en el primer comando de la instalación que el *pull* realizado es sobre el contenedor de "dkfi").

Para instalarlo, primero se ha ejecutado en la máquina con Docker el comando *docker pull dkfi/oracle-xe-10g*. A continuación, se ejecuta el comando *docker run -d -p 49160:22 -p 49161:1521 dkfi/docker-oracle-xe-10g*, que lanza el contenedor con los puertos 22 y 1251 abiertos. Una vez instalado, se ha accedido mediante el usuario 'system' (cuya contraseña es 'oracle', según la documentación) para crear un usuario destinado a manejar el almacén de datos. Para acceder, basta con utilizar el comando *sqlplus system/oracle@//192.168.99.100:49161/xe* en el *instant client* de Oracle instalado en el sistema Windows. Una vez accedido, se ha creado el usuario 'tfg' (con contraseña 'gft') y se le han otorgado los permisos necesarios, como se puede comprobar en la Figura 4.1. De esta forma, ya es posible conectarse con el usuario 'tfg'

con el comando *sqlplus tfg/gft@//192.168.99.100:49161/xe*. Nótese que 192.168.99.100 es la IP de la máquina de Docker.

```
CREATE USER tfg IDENTIFIED BY gft;
GRANT CONNECT TO tfg;
GRANT CONNECT, RESOURCE TO tfg;
GRANT CREATE SESSION TO tfg;
GRANT CREATE VIEW TO tfg;
```

Figura 4.1: Creación del usuario 'tfg'.

Tras diseñar el almacén de datos y tener acceso al sistema, se estudió cómo implementar las tablas. Los elementos que más dificultades dieron fueron los arrays, para los que se estudiaron distintas alternativas de Oracle [52]. Con todas ellas los usuarios han de utilizar PL/SQL para formular consultas que las requieran. Se estudiaron y resumieron sus características para poder realizar una decisión informada. Estas son las alternativas que se plantearon:

- Array asociativo: es un array de tipo clave-valor, siendo la clave un número o un string. Es especialmente útil para tablas de *lookup* de tamaño pequeño-medio y para tablas que contienen datos temporales. No tiene un límite máximo fijado.
- Tabla anidada: no tiene límite superior, por lo que puede aumentar de tamaño dinámicamente. Se pueden eliminar elementos de forma arbitraria, dejando los índices donde se almacenaban vacíos. Funciona bien para consultas de elementos sueltos.
- Varray: tienen un límite de elementos fijo, pero es el usuario el que, al crearlo, decide cuál es. Está almacenado como un solo objeto.

Con esta información, se decidió que la colección que más se ajustaba a las restricciones de los datos y su lógica era el varray, ya que se conoce el límite superior de todos los arrays que se van a utilizar, y es relativamente sencillo de recorrer y manejar.

El fichero completo de creación de tablas se puede consultar en el repositorio del proyecto en GitHub [53].

4.2. Diseño de los procesos ETL

El siguiente paso es crear los procesos ETL (*Extract, Transform and Load*) para limpiar, integrar y cargar los datos. Esta fase es en la que más tiempo se invirtió y

a la que más esfuerzos se dedicó, debido a las grandes diferencias existentes entre los conjuntos de datos iniciales y sus limitaciones, mencionadas en el Anexo B. Se ha tratado de hacer estos procesos reejecutables y con un diseño modular. Con ese fin se han utilizado ficheros auxiliares con información de pasos intermedios.

Se ha decidido utilizar el programa KNIME [32], ya que ya se tenía cierta experiencia con el mismo, por lo que se conocían sus capacidades y virtudes. También se han creado algunas funciones en Java con IntelliJ IDEA [33] que manejaran los datos y les dieran el formato adecuado en un par de ocasiones, como cuando se tuvieron que comprobar orden, participantes, minutos jugados y ubicación de los cambios. Esto fue necesario por la gran cantidad de datos que se estaban manejando, y cuyo procesamiento KNIME no soportaba o iba a requerir de demasiado tiempo, tiempo que se decidió mejor invertido creando las funciones.

En este apartado no se van a mostrar todos los flujos de trabajo realizados debido a su tamaño, ni se va a explicar todo lo que se ha realizado ya que en la mayoría de ocasiones se trata de manejar y transformar datos para darles el formato deseado. No obstante, se van a explicar algunos procesos, más costosos y complicados, y las soluciones que se les han dado.

El procedimiento habitual, en los casos en los que tan solo había que limpiar datos (no integrarlos), era el siguiente: mediante nodos "CSV Reader" se obtenían los datos de los conjuntos de datos, se filtraban (habitualmente) mediante nodos "Rule-based Row Filter" con expresiones regulares y se modificaban con nodos "Rule Engine", que permite dar valores en función de condiciones. Para crear los ids artificiales se utilizaba el nodo "Counter Generation", para diferencias de fechas el nodo "Math Formula", y para pequeños programas Java, el nodo "Java Snippet". Toda esta información se dividía en flujos de trabajo diferentes bien organizados y diferenciados, para no tener que cargar toda la información. También, por limpieza, se han creado multitud de metanodos, que permiten juntar partes del flujo de trabajo y mostrarlas como un solo nodo.

En cuanto a la introducción de datos, ésta se realizó casi en su totalidad en KNIME, con el nodo "Database Writer". En él, se indicaba que el driver de la base de datos era "oracle.jdbc.driver.OracleDriver" y su URL, "jdbc:oracle:thin:@//192.168.99.100:49161/xe". Sin embargo, no ha sido posible insertar desde KNIME las tablas que contienen arrays (AggFact_EquipoTemporada

y Fact_AlineacionesConocidas), ya que no permite dichas variables, de forma que se almacenaba cada elemento de cada array en una columna diferente, creando finalmente dos ficheros CSV. A partir de dichos ficheros y con expresiones regulares se crearon sendos ficheros de inserción que se lanzaron de forma manual en el almacén de datos.

La parte que más tiempo y esfuerzo requirió y que, por tanto, más detalladamente se va a explicar, fue la de lograr las correspondencias entre equipos y entre jugadores de diferentes conjuntos de datos. Puesto que se poseían hasta tres conjuntos de datos iniciales diferentes, tanto los equipos como los jugadores tenían diferentes nombres en ellos, por lo que ha sido complicado poder asegurar que el jugador de un evento es el mismo en una alineación, por ejemplo. Parece sencillo en el caso de encontrarse con "Lionel Messi" y "Leo Messi", pero no tanto en el caso de "Ander" y "Ander Herrera", menos todavía con nombres y/o apellidos más comunes.

4.2.1. Correspondencias de equipos

Se comenzó con los equipos ya que, aunque hubiera tres conjuntos de datos que corresponder, la cantidad de tuplas era menor, por lo que para marcar las pautas a seguir era preferible. Primero se correspondieron los equipos del conjunto de datos de alineaciones y los del conjunto de estadios. Para ello, se listaron todos los equipos de ambos y se compararon todos con todos (con la ayuda de un nodo "Cross Joiner") utilizando un "Java Snippet" que calculaba la distancia de Levenshtein [54]. Esta distancia mide el número de diferencias entre dos cadenas de caracteres, sumando las ediciones individuales (inserción, sustitución o supresión de un carácter) necesarias en una cadena para que sea igual que la otra. Una vez obtenidas, se daban por correspondidos los equipos con distancias nulas. Este era el primer paso, el automático. El siguiente paso era filtrar las tuplas para quedarse con las que contenían las mejores distancias para cada club. De esta forma, se seleccionaban las que, con total seguridad, se podía afirmar que eran el mismo club. Finalmente, el último paso era comparar manualmente las tuplas de cada conjunto de datos que quedaban sin corresponder para elegir las correctas. Se podría pensar que con los dos primeros pasos ya se tenían todos los equipos, pero un contraejemplo de ello es que la distancia entre "Real Zaragoza" y "Zaragoza" es mayor (5) que entre "Real Zaragoza" y "Real Cartagena" (4). Para obtener la correspondencia entre los equipos del conjunto de eventos y el de alineaciones se utilizó la misma estrategia.

4.2.2. Correspondencias de jugadores

Para las correspondencias entre jugadores de los dos conjuntos de datos principales se siguió la misma estrategia, aunque fue un proceso mucho más largo y tedioso por la gran cantidad de datos, las diferencias entre ellos y el hecho de que en el conjunto de datos con eventos los jugadores no estuvieran normalizados, ya que en cada evento aparece el nombre del jugador. Se comenzó haciendo lo mismo que en el caso anterior: calcular las distancias de Levenshtein y realizar las correspondencias automáticas, y los dos primeros pasos supervisados (elegir correspondencias manualmente pero con los filtrados en función de las distancias mencionados en la sección anterior). El siguiente paso fue similar, pero filtrando únicamente las tuplas cuyos equipos y temporadas fueran equivalentes. De esta forma, la selección supervisada era más sencilla, ya que se podía distinguir más fácilmente si dos jugadores eran el mismo comparando sus equipos (y temporadas), como se puede observar en la Figura 4.2. A continuación, para cada jugador de los eventos que quedaba sin corresponder, se creó una tupla por cada jugador de la plantilla de su equipo y temporada, para comprobar cuál de ellos podría ser, como se puede comprobar en la Figura 4.3. Estos pasos se estaban realizando con los jugadores que habían sido titulares en las cinco grandes ligas para no tratar tanta información de golpe. Por ello, el siguiente paso era similar al primero pero utilizando todos los jugadores existentes en el conjunto de datos de alineaciones. Por último, se han eliminado correspondencias de jugadores del conjunto de datos de eventos que se habían relacionado con varios de alineaciones. Esto ocurrió en el primer paso, donde se relacionaban de forma automática, ya que en el conjunto de eventos muchos nombres eran sencillos, como "Álvaro", de forma que la distancia era igual a 0 en varios casos distintos, realizando varias correspondencias con distintos jugadores. Por ello, en este último paso ha sido necesario decidir qué "Álvaro" es el correcto en la correspondencia.

nombrePartidos	nombre...	tem...	equipo...	equipo...	tem...
johny	jonny	2012/2013	Celta Vigo	Celta Vigo	2012/2013
johny	jonny	2013/2014	Celta Vigo	Celta Vigo	2013/2014
johny	jonny	2014/2015	Celta Vigo	Celta Vigo	2014/2015
johny	jonny	2015/2016	Celta Vigo	Celta Vigo	2015/2016
athanasios petsos	thanos petsos	2011/2012	1. FC Kaiser...	1. FC Kaiser...	2011/2012
athanasios petsos	thanos petsos	2012/2013	Greuther Fürth	Greuther Fürth	2012/2013
victor sanchez mata	victor sanchez	2011/2012	RCD Espanyol	RCD Espanyol	2011/2012

Figura 4.2: Ejemplo de búsqueda de correspondencias supervisada (Paso 3).

tem...	nombr...	Colu...	nombre
2014/2015	FC Augsburg	abdul baba	abdul rahman baba
2014/2015	FC Augsburg	abdul baba	alex manninger
2014/2015	FC Augsburg	abdul baba	alexander esswein
2014/2015	FC Augsburg	abdul baba	caiuby
2014/2015	FC Augsburg	abdul baba	daniel baier
2014/2015	FC Augsburg	abdul baba	dominik kohr
2014/2015	FC Augsburg	abdul baba	dong-won ji
2014/2015	FC Augsburg	abdul baba	halil altintop
2014/2015	FC Augsburg	abdul baba	jan-ingwer callse...

Figura 4.3: Ejemplo de búsqueda de correspondencias supervisada (Paso 4).

Realizando la supervisión manual de las correspondencias, se han encontrado patrones en el conjunto de datos: por ejemplo, muchos jugadores asiáticos, con nombres como "Park Ji-Sung", en el de eventos aparecían como "Ji-Sung Park", y

muchos jugadores africanos perdían el apóstrofe, como “N’Zonzi”, que aparecía como “NZonzi”. También había muchos errores relacionados con caracteres especiales, como tildes o dichos apóstrofes en otros casos, como se comenta en el Anexo B.3.

4.2.3. Otros casos

Otros casos que merecen mención en cuanto a los procesos ETL son, por ejemplo, que en general en el conjunto de datos de las alineaciones las coordenadas X e Y eran fácilmente interpretables, pero no era éste el caso de los porteros. El autor del conjunto de datos decidió darles a los porteros la posición (1,1), ubicándolos en la esquina inferior izquierda del campo. Para solucionarlo, se cambió la posición de los mismos a (5,1), de forma que aparecieran en su posición habitual: la parte inferior del campo, en el centro.

Otra dificultad fue que en el conjunto de datos de eventos, en los eventos correspondientes a los penaltis no aparecía qué jugador lo había cometido. Sin embargo, en la columna del texto con el comentario del evento sí se mencionaba. Mediante expresiones regulares se consiguieron obtener los nombres de estos jugadores, pudiendo ser añadidos a los eventos. Dichas expresiones regulares pueden leerse en el Anexo B, así como más limitaciones de los conjuntos de datos superadas en los procesos ETL.

Capítulo 5

Explotación del almacén de datos

En este capítulo se va a explicar cómo se ha utilizado el almacén de datos y cómo se podría explotar más en profundidad. En primer lugar, en la Sección 5.1 se va a demostrar la utilidad del almacén de datos sin necesidad de ningún tipo de extra, realizando diversas consultas analíticas desde distintos enfoques. En la Sección 5.2 se van a mostrar varios ejemplos de informes para un entrenador creados con Microsoft PowerBI. Para finalizar, se van a exponer los resultados obtenidos con técnicas de minería de datos, en la Sección 5.3.

5.1. Consultas de explotación

Para demostrar la utilidad del almacén de datos se han realizado diversas consultas desde diferentes enfoques. El primer enfoque es el del entrenador de un equipo y el segundo es mucho más diverso. Se pueden encontrar las consultas realizadas en el Anexo D.

Desde el enfoque del entrenador se busca optimizar el rendimiento del equipo en su conjunto, encontrando puntos débiles para evitarlos o puntos fuertes para explotarlos, con el fin último de aumentar sus probabilidades de ganar partidos. En general, se busca encontrar datos que respalden sus decisiones o ampliar su perspectiva. Algunas de las situaciones planteadas son: saber qué jugadores del equipo tienen mejor rendimiento ofensivo, encontrar zonas del campo donde el próximo rival es más débil defensivamente, saber en qué minutos el equipo recibe más goles o encontrar formaciones que no están obteniendo el rendimiento esperado.

El segundo enfoque está menos estructurado, ya que las consultas no tienen un objetivo común. Se buscan desde curiosidades que le podrían interesar a un lector habitual de la prensa deportiva (como los suplentes que han marcado más goles) hasta

algunas clasificaciones de trofeos individuales, pasando por datos objetivos que podrían ayudar a un director deportivo a rastrear el mercado en busca de los jugadores que más se adapten a las necesidades del equipo, como los mejores jugadores sub-21. También se han realizado consultas con PL/SQL [55] y con operadores de agregación [56], para ampliar conocimientos sobre los mismos.

5.2. Visualización de informes

En esta sección se van a mencionar los cuadros de mandos dinámicos que se han creado en la herramienta Microsoft PowerBI [34], y que sirven como informes para entrenadores. También se llegó a utilizar el programa DAX Studio [41] para pequeñas modificaciones de apoyo.

Se han realizado cuatro cuadros de mandos: el primero (Figura E.1) muestra estadísticas de rendimiento del equipo seleccionado en el periodo de tiempo y/o temporada seleccionados. El segundo cuadro (Figura E.2) muestra las estadísticas del partido seleccionado de un modo muy visual. El tercero (Figura E.3) muestra las alineaciones y los cambios del partido seleccionado en un diagrama de un campo de fútbol. El último (Figura E.4), presenta los datos básicos del jugador escogido, así como sus diferentes aptitudes en los atributos de juego. En el Anexo E se han explicado más detalladamente, así como su funcionamiento y su proceso de creación. Estos son una muestra del potencial del almacén de datos, por lo que también se han planteado en el anexo más ideas para posibles cuadros de mando futuros.

5.3. Minería de datos

En esta sección se van a explicar los distintos métodos para obtener alineaciones y grupos de jugadores creados con técnicas de minería de datos utilizando los datos del almacén de datos. En el Anexo F se puede encontrar información más desarrollada, además de evaluaciones de los algoritmos propuestos, trazas del programa orientado al usuario y alineaciones de ejemplo con todos los métodos para "El Clásico" de 2015. Para ello se ha utilizado RStudio [35].

5.3.1. Métodos de obtención de alineaciones

En esta sección se van a explicar los diferentes métodos que se han realizado para obtener alineaciones con técnicas de minería de datos, así como los realizados sin

estas técnicas, que se van a utilizar a modo de *baselines* para poder compararlos. En general, estos utilizan minería de reglas de asociación [57] ya que parecía la alternativa que mejor encajaba con el problema dado, teniendo grupos de jugadores del equipo en el antecedente y el resultado deseado en el consecuente de cada regla.

El primer *baseline* realizado es elegir once jugadores de la plantilla al azar. Se eligen un portero y diez jugadores de campo. Estos últimos no se han concretado por posición por la versatilidad habitual de las estructuras y las posiciones de los partidos, sus ambigüedades (un carrilero podría ser considerado un defensa adelantado o un centrocampista auxiliar) y la polivalencia de los jugadores de primer nivel. Como en los conjuntos iniciales no había datos de la posición de los jugadores, se han inferido de las coordenadas en las que han jugado. Por ejemplo, un jugador puede considerarse portero si ha jugado un cierto porcentaje de partidos en la coordenada $Y = 1$.

El segundo *baseline* elige los once jugadores (un portero y diez de campo) más utilizados por el entrenador a lo largo de la temporada, hasta el partido a analizar. En caso de no existir (en la primera jornada), se completa con el *baseline* al azar.

El tercer *baseline* elige, mediante minería de reglas de asociación [57], los once jugadores que más victorias han conseguido hasta el momento en la misma temporada. Utiliza el mismo algoritmo que se explicará en el siguiente párrafo, pero solo hace uso de las reglas cuya longitud del antecedente es 1. Se considera *baseline* porque se podría haber implementado sin técnicas de minería de datos.

Los dos primeros métodos (mejor resultado con la misma temporada y mejor resultado con todas las temporadas) son idénticos, excepto por los datos que utilizan para su fin. Si el parámetro *periodoDatos* es "MISMA", se utilizarán los partidos del equipo anteriores pero de la misma temporada para obtener las reglas de asociación. En cambio, si es "TODAS", se utilizarán además los partidos del equipo en todas las temporadas anteriores. Ambos tratan de encontrar a los once jugadores con más victorias gracias al uso del algoritmo a priori de reglas de asociación [58] (ordenando las obtenidas de mayor a menor *lift* [59]). A diferencia del *baseline* de más victorias, estos sí tratan de encontrar patrones en los datos, buscando reglas de asociación de longitud variable y mayor que 2, que aúne a varios jugadores para incluirlos a todos ellos en la alineación. En caso de no poder completarlo, se intenta completar con reglas de asociación de partidos empatados. Si tampoco se consigue (lo que significa que el equipo todavía no ha ganado ni empatado esa temporada), se completa con los

baselines, en orden de mayor a menor precisión, con el objetivo de nunca sugerirle al usuario un once incompleto.

El último método implementado también trata de maximizar la probabilidad de victoria, pero en este caso lo hace mediante una regresión logística [60], para explorar otras técnicas, a pesar de que con tan pocos datos no funciona tan bien como los anteriores. En el Anexo F.3 se explica por qué se han utilizado (por defecto) tan solo las victorias. Se ha descartado utilizar los datos de todas las temporadas porque los jugadores que llevan más tiempo en el equipo ganarían importancia, por lo que tendrían más probabilidades de resultar elegidos, adulterando los resultados. Para obtener la alineación, se ha creado un *data frame* [61] con una columna RESULTADO, con un valor numérico entre 0 y 1 (usando solo las victorias siempre es 1), y un número de columnas igual al tamaño de la plantilla del equipo en la temporada. Estas columnas, llamadas JUEGAX (siendo X el índice de la lista "participantes" al que recurrir posteriormente para obtener el id del jugador), indican con un valor booleano si han jugado en el partido tratado. De esta forma se crea el modelo y se crea la alineación con los once jugadores (un portero y diez de campo) con mayor coeficiente. En caso de no obtener once, se recurre al *baseline* de jugadores más usados para completar la alineación.

Por otra parte, y para utilizar otro objetivo distinto a obtener victorias, se han creado métodos mediante minería de reglas que encuentran grupos de delanteros con los que el equipo marque *muchos* goles y grupos de defensas con los que el equipo reciba *pocos*. *Muchos* y *pocos* son parámetros modificables por el administrador que por defecto son 2 o más y 1 o menos, respectivamente. En cuanto a qué jugadores pueden considerarse defensas y delanteros, esto se deduce de forma similar a los porteros. Los distintos umbrales para los porcentajes de partidos jugados en posición defensiva/atacante respecto al total de partidos jugados también son parámetros modificables. Es posible que un jugador pueda ejercer en más de una línea del campo, algo totalmente loable y que de hecho es preferible ya que no restringe al mismo en los análisis. Otro parámetro es el periodo de datos a utilizar (misma temporada o todas). En el programa del usuario, detallado en el Anexo F.1, además de poder obtener una alineación con el método de mejor resultado con la misma temporada, se le da la opción al usuario de obtener grupos de jugadores, e incluso de obtener una alineación a partir de los sugeridos.

5.3.2. Evaluación del rendimiento

En esta sección se evalúan todos los métodos creados. Cabe destacar que en este problema, a diferencia de otros casos de minería de datos, no es posible obtener una solución y afirmar fácilmente que "es mejor" que otra, ya que no hay forma posible de saber qué habría ocurrido en un partido real con la alineación propuesta. Por ello, solo se ha podido comparar con la alineación real, sin realmente saber si ésta es mejor o peor que la propuesta. Los métodos de cuantificación de diferencias entre alineaciones son dos:

- Método perfecto: Calcula el número de jugadores diferentes entre la alineación obtenida y la real.
- Método no perfecto: Calcula la diferencia en función del resultado y del número de jugadores diferentes entre la alineación obtenida y la real. De esta forma, se tiene en cuenta que la alineación real no tuvo por qué ser óptima, reduciendo el número de diferencias a la mitad si el once real empató y a un tercio si perdió.

A continuación se presentan las gráficas resumidas de los resultados de las pruebas realizadas. En ellas se muestran los errores absolutos medios, los errores cuadráticos medios y los tiempos medios de todos los métodos y *baselines*. En la tabla 5.1 se pueden comprobar los métodos y *baselines* a los que se refiere cada acrónimo.

Método/*Baseline*	Acrónimo
Al azar	AZ
Más usados	MU
Más victorias	MV
Mejor resultado (con misma temporada)	MRM
Mejor resultado (con todas las temporadas)	MRT
Regresión logística	RL

Tabla 5.1: Acrónimos de los métodos de obtención de alineaciones.

De estas gráficas se puede concluir que los métodos mejoran a los *baselines* pero no de forma sustancial. Esto se podría explicar con el "conservadurismo" de los entrenadores, que mantienen un once más o menos fijo a lo largo de toda la temporada, especialmente si logran ganar con él, variando muy poco sus alineaciones. Es por ello que los *baselines* de más usados y con más victorias funcionan bastante bien, incluso mejor que el método que utiliza una regresión logística, dado que el modelo se crea con una cantidad muy limitada de datos, por lo que no funciona todo lo bien que podría. Además, los métodos de mejores resultados son los que más tiempo necesitan para

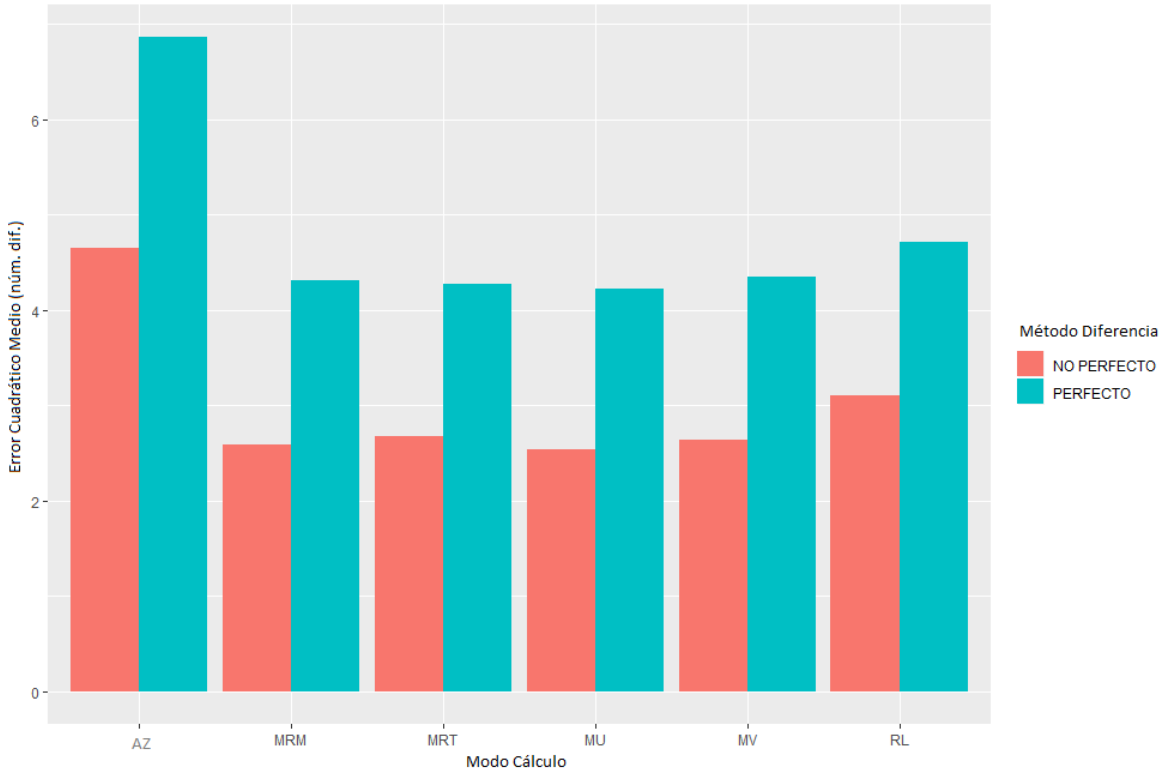

Figura 5.1: Errores cuadráticos medios.

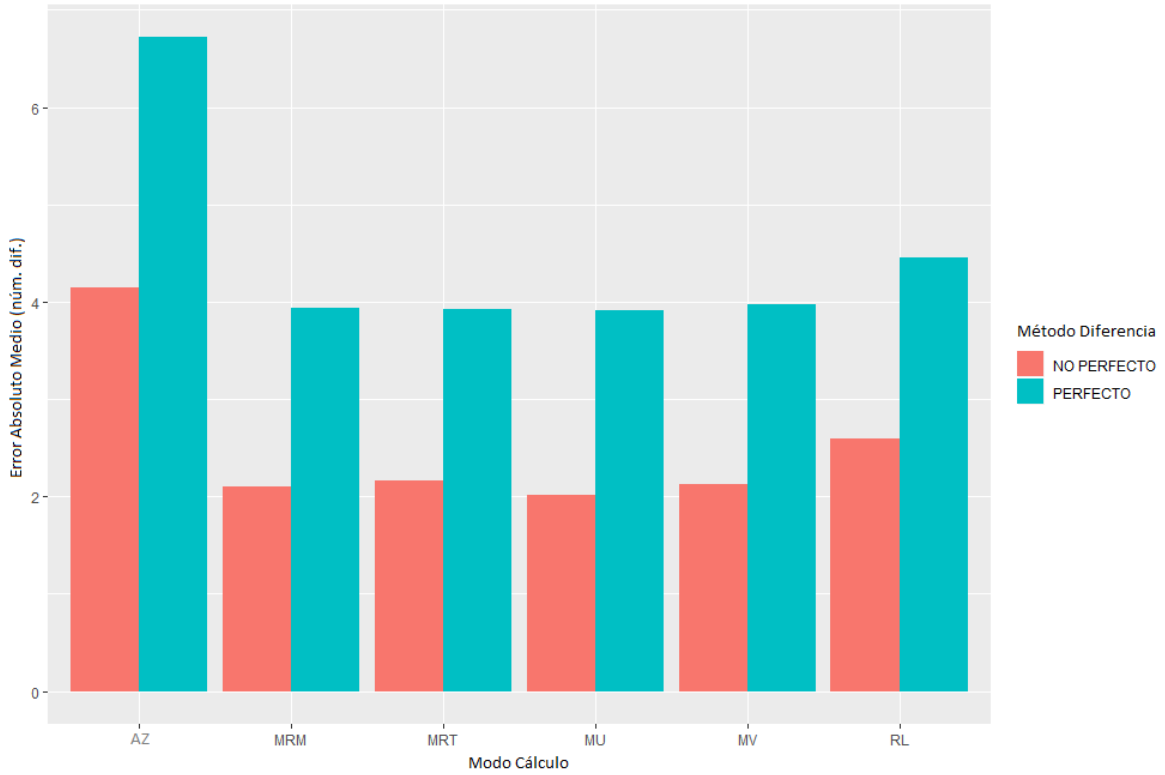

Figura 5.2: Errores absolutos medios.

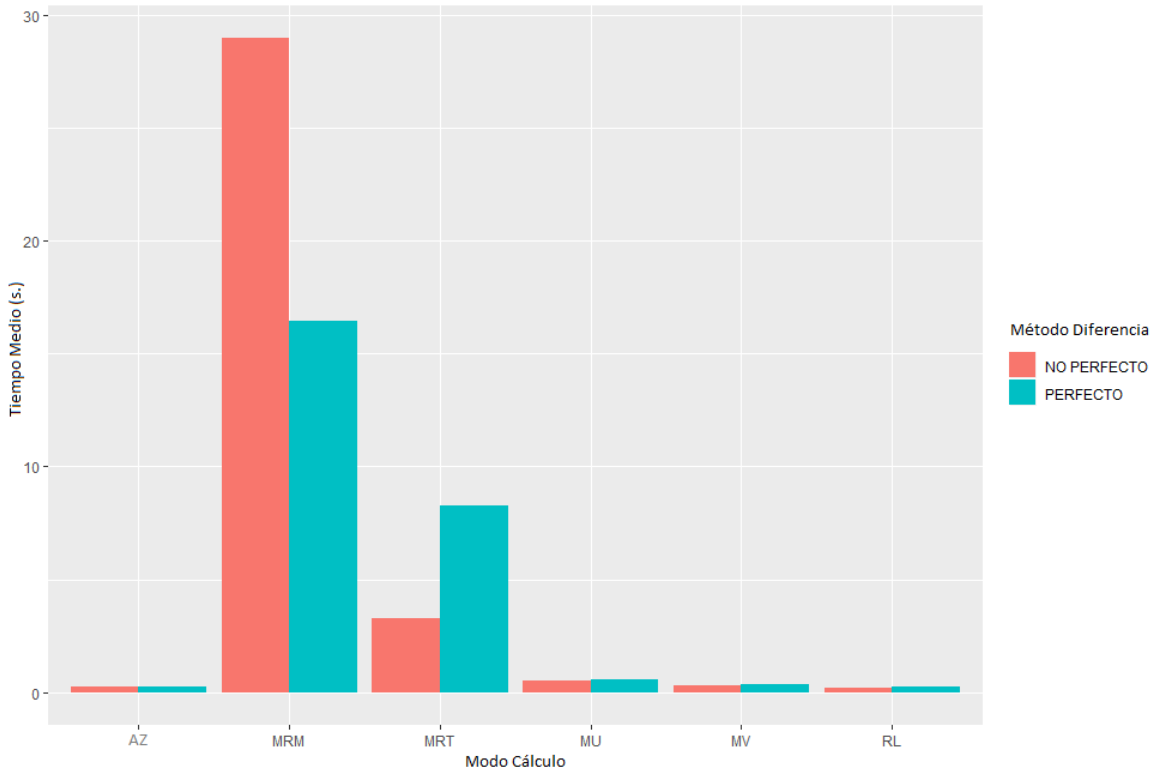

Figura 5.3: Tiempos medios (en segundos).

obtener una alineación (Figura 5.3). Por norma general esto no debería suponer un problema, ya que esta herramienta no está pensada para utilizarse en tiempo real, pero en caso de requerirlo, podría ser preferible optar por el método con la regresión lineal o el de jugadores con más victorias. Para más información y explicaciones más extensas, véase el Anexo F.2.

Capítulo 6

Conclusiones y trabajo futuro

En este último capítulo se desarrollan las conclusiones del proyecto y sus posibilidades de cara al futuro. En la Sección 6.1 se detallan las conclusiones del trabajo, en la Sección 6.2 las conclusiones personales, con lecciones aprendidas y dificultades encontradas, y en la Sección 6.3 el trabajo futuro.

6.1. Conclusiones del trabajo

Está empezando una nueva etapa en el deporte con la inclusión de la ciencia de datos y, ya que los propios deportes generan una ingente cantidad de datos, hay muchas posibles vías por las que se pueden aprovechar. Con este proyecto se ha intentado dar un primer paso hacia este mundo, creando una herramienta práctica y ampliable, que se puede recrear utilizando los ficheros ubicados en el repositorio de GitHub del proyecto [53]. La decisión de hacerlo público se sustenta en que se cree que el almacén de datos y los informes, pruebas y algoritmos desarrollados pueden ser de interés no solo para realizar mejoras, sino también desde una perspectiva docente, en asignaturas tales como "Almacenes y minería de datos" [62] o "Bases de datos 2" [63]. También para curiosos que quieran probar alguna de sus funciones. La herramienta consta de un almacén de datos, que se ha diseñado, implementado y poblado con datos reales, que permite multitud de formas de explotación, como consultas analíticas o informes orientados a humanos. Además, se han creado técnicas propias que permiten sugerir alineaciones para partidos concretos, utilizando minería de reglas de asociación y regresiones logísticas.

Se ha comprobado que no es sencillo formar una alineación que asegure un gran rendimiento debido a la impredecibilidad de este deporte y al gran número de variables que influyen. Pese a ello, ha quedado claro que los entrenadores no suelen atreverse a modificar mucho sus alineaciones, manteniendo una base fija. Otra de las conclusiones

es que en el fútbol, pese a esta impredecibilidad, también es posible tomar decisiones informadas en función de los datos, si son expertos los que analizan estos datos con criterio.

6.2. Conclusiones personales

Personalmente, crear todo un proyecto desde cero ha supuesto una satisfacción y especialmente un reto, ya que las prácticas de las asignaturas del grado eran problemas más sencillos y habitualmente no formaban parte de un proyecto conexo. Tener en mente unos objetivos principales durante todo el transcurso del proyecto y estar centrado completamente en ellos derivaba en una sensación diferente a la hora de trabajar. Además, se han ampliado considerablemente los conocimientos previos en materia de almacenes de datos, minería de datos, procesos ETL, etc., que no pasaban de la mera teoría y pequeñas partes más prácticas. En cuanto al tema elegido, se escogió el fútbol porque es un tema que se conoce en profundidad (sin llegar a ser un experto) y para el que está empezando a cobrar importancia el *data science* [64], como se ha comentado anteriormente. Por ello, este trabajo podría ayudar a abrir salidas profesionales atractivas en el sector.

En cuanto a las dificultades encontradas, las principales han sido conseguir conjuntos de datos que cumplieran al menos unos requisitos mínimos de calidad y cantidad de datos (tanto que ha sido necesario integrar tres conjuntos diferentes), conseguir integrar todos esos datos y aplicar técnicas de minería de datos hasta obtener resultados.

El proyecto ha resultado enriquecedor y se han aprendido varias lecciones. Entre ellas, llevar una organización con un proyecto de un tamaño mayor a los realizados en el grado, conocer la existencia de herramientas y plataformas como Kaggle o LaTeX, saber desenvolverse mejor con otras como R o KNIME, ampliar conocimientos teóricos y prácticos sobre almacenes de datos y superar la dificultad de trabajar con datos reales.

6.3. Trabajo futuro

En cuanto al trabajo futuro, este proyecto puede servir como base para muchos otros. Se han planteado algunas mejoras y otras perspectivas sin necesidad de modificar nada e investigaciones alternativas en caso de poder mejorar el almacén de datos.

Sin modificarlo, otras formas de explotar el almacén serían tratar de identificar otro tipo de patrones en los datos diferentes a alineaciones, tratar de descubrir jugadores que fichar o incluso se podría utilizar en cursos de entrenadores como una especie de simulador. Obviamente también sirve para mantener datos históricos, que podrían servir para estadísticas o curiosidades, de una forma similar a los *tweets* de "Mister Chip", una de las referencias del periodismo deportivo español actual [65].

Sin embargo, modificándolo es donde podría mostrar todo su potencial: integrando una fuente de datos fiable y en tiempo real, sería posible recomendar alineaciones de partidos venideros, o incluso recomendar cambios en el transcurso del partido. Si las fuentes de datos incluyeran ligas no profesionales, esta herramienta podría resultar mucho más beneficiosa para el equipo que la utilizara, ya que presumiblemente los demás equipos no utilizan estas técnicas modernas. También podría ampliarse el almacén de datos, adaptándolo a un equipo concreto, con información mucho más detallada de sus jugadores (física, psicológica, preferencias del jugador, etc.). Si las hipotéticas fuentes de datos contuvieran más tipos de datos se podría aumentar la precisión de técnicas ya realizadas o plantear otro tipo de objetivos, como utilizar la herramienta para realizar seguimiento de posibles fichajes, sirviendo de herramienta para un director deportivo. Otros tipos de datos de los que se hubiera deseado disponer, como información de lesiones o económica, se pueden encontrar en el Anexo B.3, donde se comenta qué se podría hacer con algunos de ellos.

Bibliografía

[1] How data, not people, call the shots in Denmark. https://thecorrespondent.com/2607/how-data-not-people-call-the-shots-in-denmark/230219386155-d2948861. Accedido por última vez el 8 de noviembre de 2019.

[2] Moneyball: la película de las estadísticas y Sabermetría. https://medium.com/qu4nt/moneyball-y-sabermetria-627ab92550b7. Accedido por última vez el 8 de noviembre de 2019.

[3] Esta zamorana experta en 'big data' es el gran secreto de la selección española de 'basket'. https://www.elconfidencial.com/tecnologia/2019-07-10/espanola-big-data-seleccion-espanola-baloncesto-femenina_2114567/. Accedido por última vez el 12 de noviembre de 2019.

[4] VAR: ¿Qué es y cómo funciona? https://www.mundodeportivo.com/futbol/20190302/421283672203/var-que-es-como-funciona-videoarbitraje.html. Accedido por última vez el 8 de noviembre de 2019.

[5] El matemático que te dice cómo no fallar un penalti. http://prensa.unizar.es/noticias/1911/191114_z0_mundo29.pdf. Accedido por última vez el 14 de noviembre de 2019.

[6] How Newcastle United got a sports science edge: Big data, blood tests and brilliant training plans. https://www.chroniclelive.co.uk/sport/football/football-news/blood-tests-big-data-rafa-16224870. Accedido por última vez el 12 de noviembre de 2019.

[7] La reinvención de Monchi, el mago de los fichajes: "Obviar el 'Big Data' es anacrónico". https://www.elconfidencial.com/deportes/futbol/2019-10-17/entrevista-monchi-sevilla-big-data-443_2278023/. Accedido por última vez el 8 de noviembre de 2019.

[8] 'Big Data' en el fútbol — Visionarios — El País Semanal. https://www.youtube.com/watch?v=FX9X9IJop5A. Accedido por última vez el 8 de noviembre de 2019.

[9] Opta Sports. https://www.optasports.com/. Accedido por última vez el 8 de noviembre de 2019.

[10] Wyscout. https://wyscout.com/. Accedido por última vez el 8 de noviembre de 2019.

[11] SAP Sports One. https://www.sap.com/products/sports-one.html. Accedido por última vez el 8 de noviembre de 2019.

[12] Driblab. https://driblab.com/es/inicio/. Accedido por última vez el 8 de noviembre de 2019.

[13] Gantt. https://www.gantt.com/. Accedido por última vez el 13 de noviembre de 2019.

[14] Excel — Cómo hacer un diagrama de Gantt o cronograma utilizando los gráficos. https://saberprogramas.com/excel-como-hacer-un-diagrama-gantt-en-excel/. Accedido por última vez el 13 de noviembre de 2019.

[15] Todoist. https://todoist.com/es. Accedido por última vez el 13 de noviembre de 2019.

[16] Google Drive. https://www.google.com/intl/es_ALL/drive/. Accedido por última vez el 13 de noviembre de 2019.

[17] Microsoft Excel. https://products.office.com/es-es/excel. Accedido por última vez el 13 de noviembre de 2019.

[18] D. Memmert R. Rein. Big data and tactical analysis in elite soccer: future challenges and opportunities for sports science. *SpringerPlus*, 5, August 2016.

[19] N. Nourafza M. Haghighat, H. Rastegari. A review of data mining techniques for result prediction in sports. *ACSIJ Advances in Computer Science: an International Journal*, 2, November 2013.

[20] Paola Zuccolotto, Maurizio Carpita, Marco Sandri, and Anna Simonetto. Discovering the drivers of football match outcomes with data mining. *Quality Technology and Quantitative Management*, 12:537–553, January 2015.

[21] D. Pedreschi F. Giannotti M. Malvaldi P. Cintia, L. Pappalardo. The harsh rule of the goals: data-driven performance indicators for football teams. *2015 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, pages 1–10, October 2015.

[22] Rabah Al-Shboul, Tahir Syed, Jamshed Memon, and Furqan Khan. Automated Player Selection for a Sports Team using Competitive Neural Networks. *International Journal of Advanced Computer Science and Applications*, 8(8), 2017.

[23] Babatunde Buraimo, David Forrest, and Robert Simmons. The 12th man?: Refereeing Bias in English and German Soccer. *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, 173(2):431–449, 2010.

[24] Sublime Text. `https://www.sublimetext.com/`. Accedido por última vez el 6 de noviembre de 2019.

[25] Overleaf. `https://www.overleaf.com/`. Accedido por última vez el 6 de noviembre de 2019.

[26] Kaggle. `https://www.kaggle.com/`. Accedido por última vez el 6 de noviembre de 2019.

[27] Sergio Ilarri. DBDAp: Una herramienta de apoyo a la docencia de bases de datos y almacenes de datos. *ReVisión*, 10(1), January 2017. 29–53.

[28] Oracle Express Edition. `https://www.oracle.com/database/technologies/appdev/xe.html`. Accedido por última vez el 6 de noviembre de 2019.

[29] Docker. `https://www.docker.com/`. Accedido por última vez el 6 de noviembre de 2019.

[30] Docker Hub. `https://hub.docker.com/`. Accedido por última vez el 6 de noviembre de 2019.

[31] Oracle VirtualBox. `https://www.virtualbox.org/`. Accedido por última vez el 8 de noviembre de 2019.

[32] KNIME. `https://www.knime.com/`. Accedido por última vez el 8 de noviembre de 2019.

[33] IntelliJ IDEA. `https://www.jetbrains.com/idea/`. Accedido por última vez el 6 de noviembre de 2019.

[34] Microsoft PowerBI. https://powerbi.microsoft.com/es-es/. Accedido por última vez el 6 de noviembre de 2019.

[35] RStudio. https://rstudio.com/products/rstudio/. Accedido por última vez el 6 de noviembre de 2019.

[36] Tableau. https://www.tableau.com/es-es. Accedido por última vez el 6 de noviembre de 2019.

[37] R. https://www.r-project.org/. Accedido por última vez el 6 de noviembre de 2019.

[38] Weka. https://www.cs.waikato.ac.nz/ml/weka/. Accedido por última vez el 6 de noviembre de 2019.

[39] Wikidata. https://query.wikidata.org/. Accedido por última vez el 6 de noviembre de 2019.

[40] SQLiteStudio. https://sqlitestudio.pl/. Accedido por última vez el 6 de noviembre de 2019.

[41] DAX Studio. https://daxstudio.org/. Accedido por última vez el 6 de noviembre de 2019.

[42] Conceptos básicos de ayuda de CRISP-DM — IBM. https://www.ibm.com/support/knowledgecenter/es/SS3RA7_sub/modeler_crispdm_ddita/clementine/crisp_help/crisp_overview.html. Accedido por última vez el 12 de noviembre de 2019.

[43] Dimensional Modeling Techniques — Kimball Group. https://www.kimballgroup.com/data-warehouse-business-intelligence-resources/kimball-techniques/dimensional-modeling-techniques/. Accedido por última vez el 12 de noviembre de 2019.

[44] Inmon or Kimball: Which approach is suitable for your data warehouse? https://www.computerweekly.com/tip/Inmon-or-Kimball-Which-approach-is-suitable-for-your-data-warehouse. Accedido por última vez el 12 de noviembre de 2019.

[45] European Soccer Database — Kaggle. https://www.kaggle.com/hugomathien/soccer. Accedido por última vez el 8 de noviembre de 2019.

[46] Football Events — Kaggle. https://www.kaggle.com/secareanualin/football-events. Accedido por última vez el 8 de noviembre de 2019.

[47] El Rayo Majadahonda jugará en el Wanda Metropolitano. https://as.com/futbol/2018/07/11/segunda/1531326084_854729.html. Accedido por última vez el 20 de noviembre de 2019.

[48] Data Warehouse Design Techniques — Rapidly Changing Dimensions. https://www.nuwavesolutions.com/rapidly-changing-dimensions/. Accedido por última vez el 12 de noviembre de 2019.

[49] Factless Fact Table. https://www.jamesserra.com/archive/2011/12/factless-fact-table/. Accedido por última vez el 19 de noviembre de 2019.

[50] Imagen de Docker Oracle 10g Express Edition utilizada. https://hub.docker.com/r/dkfi/docker-oracle-xe-10g/. Accedido por última vez el 8 de noviembre de 2019.

[51] Imagen de Docker Oracle 10g Express Edition no utilizada. https://hub.docker.com/r/sandersliu/docker-oracle-xe-10g/. Accedido por última vez el 20 de septiembre de 2019.

[52] Colecciones de Oracle. https://docs.oracle.com/cd/B10501_01/appdev.920/a96624/05_colls.htm. Accedido por última vez el 8 de noviembre de 2019.

[53] Repositorio del proyecto en GitHub. https://github.com/opotrony/TFG. Accedido por última vez el 7 de noviembre de 2019.

[54] Código fuente del cálculo de la Distancia de Levenshtein. https://en.wikipedia.org/wiki/Levenshtein_distance#Computing_Levenshtein_distance. Accedido por última vez el 8 de noviembre de 2019.

[55] PL/SQL para desarrolladores. https://www.oracle.com/es/database/technologies/appdev/plsql.html. Accedido por última vez el 16 de noviembre de 2019.

[56] GROUPING SETS Aggregate. https://www.vertica.com/docs/9.2.x/HTML/Content/Authoring/SQLReferenceManual/Statements/SELECT/GROUPINGSETaggregate.htm. Accedido por última vez el 16 de noviembre de 2019.

[57] Association Rule Mining. https://towardsdatascience.com/association-rule-mining-be4122fc1793. Accedido por última vez el 16 de noviembre de 2019.

[58] Apriori Algorithm In Data Mining: Implementation With Examples. https://www.softwaretestinghelp.com/apriori-algorithm/. Accedido por última vez el 16 de noviembre de 2019.

[59] Complete guide to Association Rules. https://towardsdatascience.com/association-rules-2-aa9a77241654. Accedido por última vez el 18 de noviembre de 2019.

[60] Logistic Regression For Dummies: A Detailed Explanation. https://towardsdatascience.com/logistic-regression-for-dummies-a-detailed-explanation-9597f76edf46. Accedido por última vez el 16 de noviembre de 2019.

[61] Data Frames. https://bookdown.org/jboscomendoza/r-principiantes4/data-frames.html. Accedido por última vez el 16 de noviembre de 2019.

[62] Almacenes y minería de datos — Universidad de Zaragoza. https://estudios.unizar.es/estudio/asignatura?anyo_academico=2019&asignatura_id=30253&estudio_id=20190148&centro_id=110&plan_id_nk=439. Accedido por última vez el 21 de noviembre de 2019.

[63] Bases de datos 2 — Universidad de Zaragoza. https://estudios.unizar.es/estudio/asignatura?anyo_academico=2019&asignatura_id=30250&estudio_id=20190148&centro_id=110&plan_id_nk=439. Accedido por última vez el 21 de noviembre de 2019.

[64] What is Data Science? https://datascience.berkeley.edu/about/what-is-data-science/. Accedido por última vez el 20 de noviembre de 2019.

[65] Perfil de Mister Chip en Twitter. https://twitter.com/2010MisterChip. Accedido por última vez el 17 de noviembre de 2019.

[66] draw.io — Flowchart maker. https://www.draw.io/. Accedido por última vez el 16 de noviembre de 2019.

[67] Fantasy Premier League. https://fantasy.premierleague.com/help/rules. Accedido por última vez el 16 de noviembre de 2019.

[68] Enterprise Data Warehouse Bus Architecture. https://www.kimballgroup.com/data-warehouse-business-intelligence-resources/kimball-techniques/kimball-data-warehouse-bus-architecture/. Accedido por última vez el 16 de noviembre de 2019.

[69] Nulls in Fact Tables. https://www.kimballgroup.com/data-warehouse-business-intelligence-resources/kimball-techniques/dimensional-modeling-techniques/fact-table-null/. Accedido por última vez el 16 de noviembre de 2019.

[70] Null Attributes in Dimensions. https://www.kimballgroup.com/data-warehouse-business-intelligence-resources/kimball-techniques/dimensional-modeling-techniques/null-dimension-attribute/. Accedido por última vez el 16 de noviembre de 2019.

[71] Imagen de campo de fútbol. https://upload.wikimedia.org/wikipedia/commons/f/f3/Football_field_105x68.PNG. Accedido por última vez el 9 de noviembre de 2019.

[72] Infographic Designer — PowerBI Marketplace. https://appsource.microsoft.com/en-us/product/power-bi-visuals/WA104380898?tab=Overview. Accedido por última vez el 9 de noviembre de 2019.

[73] Tornado Chart — PowerBI Marketplace. https://appsource.microsoft.com/en-us/product/power-bi-visuals/WA104380768?tab=Overview. Accedido por última vez el 9 de noviembre de 2019.

[74] Enhanced Scatter — PowerBI Marketplace. https://appsource.microsoft.com/en-us/product/power-bi-visuals/WA104380762?tab=Overview. Accedido por última vez el 16 de noviembre de 2019.

[75] Radar Chart — PowerBI Marketplace. https://appsource.microsoft.com/en-us/product/power-bi-visuals/WA104380771?tab=Overview. Accedido por última vez el 9 de noviembre de 2019.

[76] El PSV mete un 10-0 histórico al Feyenoord. https://www.20minutos.es/deportes/noticia/psv-feyenord-goleada-852620/0/. Accedido por última vez el 8 de noviembre de 2019.

[77] Troyes 0 - 9 Paris St-Germain. https://www.bbc.com/sport/football/35749229. Accedido por última vez el 8 de noviembre de 2019.

[78] Akaike's Information Criterion: Definition, Formulas... https://www.statisticshowto.datasciencecentral.com/akaikes-information-criterion/. Accedido por última vez el 16 de noviembre de 2019.

Lista de Figuras

Lista de Tablas

Anexos

Anexo A

Tiempo dedicado

En este anexo se desglosan las horas dedicadas semanalmente a cada apartado durante todo el desarrollo del proyecto, cuyo resumen se muestra en la Sección 1.3.

El número de horas semanales realizadas se puede ver en las tablas A.1 y A.2. En ambas las columnas significan, de izquierda a derecha, el día de inicio de la semana indicada, el apartado de obtención y análisis de datos, el apartado de diseño del almacén de datos y el de implementación de éste, los apartados de consultas, visualización y minería de datos y el tiempo ocupado en realizar la memoria. Todo está medido en horas.

Semana	Obt	Dis	Imp	Con	Vis	Min	Mem
19/11/2018	4						
26/11/2018	1						
03/12/2018	1						
10/12/2018	1						
17/12/2018	3						
24/12/2018							
31/12/2018							
07/01/2019							
14/01/2019							
21/01/2019							
28/01/2019							
04/02/2019							
11/02/2019		4					

Tabla A.1: Tiempo dedicado dividido en semanas y apartados (1 de 2).

Semana	Obt	Dis	Imp	Con	Vis	Min	Mem
18/02/2019		3					
25/02/2019							
04/03/2019		2					
11/03/2019		3					
18/03/2019			5				
25/03/2019							
01/04/2019							
08/04/2019							
15/04/2019		2	15				
22/04/2019							
29/04/2019							
06/05/2019							
13/05/2019							
20/05/2019			15				
27/05/2019			25				
03/06/2019			25				
10/06/2019		4	25				
17/06/2019			8				
24/06/2019			30				
01/07/2019			30				
08/07/2019			30				
15/07/2019			25	10			
22/07/2019				30			
29/07/2019				10	25		
05/08/2019					20		5
12/08/2019							3
19/08/2019					10		
26/08/2019					5		8
02/09/2019							2
09/09/2019							
16/09/2019							5
23/09/2019							5
30/09/2019							10
07/10/2019						10	10
14/10/2019						20	
21/10/2019						30	
28/10/2019						25	
04/11/2019						10	15
11/11/2019							8
18/11/2019							8

Tabla A.2: Tiempo dedicado dividido en semanas y apartados (2 de 2).

Anexo B

Estudio previo de los conjuntos de datos seleccionados

En el Anexo B.1 se realiza un estudio preliminar de los conjuntos de datos iniciales, analizando los datos disponibles. Después, en el Anexo B.2 se muestran y comentan sus esquemas relacionales. Finalmente, en el Anexo B.3 se comentan sus limitaciones y se enumeran otros tipos de datos que se hubieran deseado tener.

B.1. Análisis preliminar

En esta sección se va a explicar el estudio realizado sobre los conjuntos de datos elegidos finalmente: "European Soccer Database" [45] y "Football Events" [46], así como el fichero CSV de estadios obtenido de Wikidata.

Para comenzar con el estudio, se ha realizado un recuento del número de partidos por liga del primer conjunto de datos (Figura B.1). Cabe destacar que en Alemania (cuyo league_id es 7809) juegan 18 equipos, a diferencia de las demás, donde juegan 20 equipos. Se contabilizan cuatro temporadas, con partidos de ida y vuelta.

```
> count(match.data,'league_id')
  league_id freq
1      1729 3040
2      4769 3040
3      7809 2448
4     10257 3017
5     21518 3040
```

Figura B.1: Recuento de partidos por liga.

Después, para contabilizar los valores nulos de los conjuntos de datos, se ha creado una función (Figura B.2) que muestre el porcentaje de nulos de cada atributo y el

número total de los mismos, si existen, como se puede ver en la Figura B.3 y en la Figura B.4.

```
# function porcentajeDeNulos - muestra por pantalla el
#       porcentaje de nulos de las columnas del data frame d,
#       si es distinto de 0.
porcentajeDeNulos <- function(d){
  cat(sprintf("\nAnalizando nulos de %s\n",
        deparse(substitute(d))))
  y <- nrow(d)
  z <- colnames(d, do.NULL = FALSE)
  for(i in 1:length(z)){
    x <- sum(is.na(d[i]))
    if (x > 0){
      cat(sprintf("---> %s: %s %%(%s de %s)\n", z[i],
            round(x*100/y, digits = 4), x, y))
    }
  }
}
```

Figura B.2: Función porcentajeDeNulos.

Como se puede observar, el conjunto de datos de eventos es el que tiene más nulos. Sin embargo, esto no es un problema, ya que se debe a que utiliza unos atributos u otros en función del tipo de evento, siendo los demás atributos nulos. Los atributos que contienen nulos que más consecuencias podrían acarrear son los de jugadores del conjunto de datos de partidos. No obstante, es un porcentaje bajo, no pasando del 0.5 %.

El último apartado de este estudio preliminar es la búsqueda de datos anómalos. Se han buscado en distintos atributos de los conjuntos de datos, sin encontrar ninguno. Las jornadas varían entre 1 y 38 y no había ningún número de goles negativo. En el primer conjunto de datos, el máximo número de goles locales es 10 y de visitante, 9. Se han comprobado ambos partidos y los datos son correctos: PSV 10 - 0 Feyenoord [76] y Troyes 0 - 9 PSG [77]. En el conjunto de datos de eventos también aparecía el 0-9, no así el 10-0 porque no tiene información de la temporada en la que ocurrió. También se han realizado histogramas de las valoraciones generales de jugadores (Figura B.5), sus potenciales (Figura B.6), sus alturas (Figura B.7) y sus pesos (Figura B.8) y todos ellos son razonables.

```
Analizando nulos de match.data
---> away_player_X11: 0.0343 % (5 de 14585)
---> away_player_Y11: 0.0343 % (5 de 14585)
---> home_player_1: 0.2605 % (38 de 14585)
---> home_player_2: 0.3222 % (47 de 14585)
---> home_player_3: 0.2743 % (40 de 14585)
---> home_player_4: 0.3017 % (44 de 14585)
---> home_player_5: 0.2605 % (38 de 14585)
---> home_player_6: 0.2331 % (34 de 14585)
---> home_player_7: 0.3565 % (52 de 14585)
---> home_player_8: 0.3017 % (44 de 14585)
---> home_player_9: 0.2331 % (34 de 14585)
---> home_player_10: 0.5005 % (73 de 14585)
---> home_player_11: 0.4251 % (62 de 14585)
---> away_player_1: 0.1988 % (29 de 14585)
---> away_player_2: 0.336 % (49 de 14585)
---> away_player_3: 0.3154 % (46 de 14585)
---> away_player_4: 0.3154 % (46 de 14585)
---> away_player_5: 0.3017 % (44 de 14585)
---> away_player_6: 0.3085 % (45 de 14585)
---> away_player_7: 0.2948 % (43 de 14585)
---> away_player_8: 0.3771 % (55 de 14585)
---> away_player_9: 0.3222 % (47 de 14585)
---> away_player_10: 0.4251 % (62 de 14585)
---> away_player_11: 0.4937 % (72 de 14585)

Analizando nulos de country.data

Analizando nulos de league.data

Analizando nulos de player.data

Analizando nulos de playerAtt.data
---> overall_rating: 0.4544 % (836 de 183978)
---> potential: 0.4544 % (836 de 183978)
---> crossing: 0.4544 % (836 de 183978)
---> finishing: 0.4544 % (836 de 183978)
---> heading_accuracy: 0.4544 % (836 de 183978)
---> short_passing: 0.4544 % (836 de 183978)
---> volleys: 1.4746 % (2713 de 183978)
---> dribbling: 0.4544 % (836 de 183978)
---> curve: 1.4746 % (2713 de 183978)
---> free_kick_accuracy: 0.4544 % (836 de 183978)
---> long_passing: 0.4544 % (836 de 183978)
---> ball_control: 0.4544 % (836 de 183978)
---> acceleration: 0.4544 % (836 de 183978)
---> sprint_speed: 0.4544 % (836 de 183978)
---> agility: 1.4746 % (2713 de 183978)
---> reactions: 0.4544 % (836 de 183978)
---> balance: 1.4746 % (2713 de 183978)
---> shot_power: 0.4544 % (836 de 183978)
---> jumping: 1.4746 % (2713 de 183978)
---> stamina: 0.4544 % (836 de 183978)
---> strength: 0.4544 % (836 de 183978)
---> long_shots: 0.4544 % (836 de 183978)
---> aggression: 0.4544 % (836 de 183978)
---> interceptions: 0.4544 % (836 de 183978)
---> positioning: 0.4544 % (836 de 183978)
```

Figura B.3: Resultado del estudio de valores nulos (1 de 2).

```
---> vision: 1.4746 % (2713 de 183978)
---> penalties: 0.4544 % (836 de 183978)
---> marking: 0.4544 % (836 de 183978)
---> standing_tackle: 0.4544 % (836 de 183978)
---> sliding_tackle: 1.4746 % (2713 de 183978)
---> gk_diving: 0.4544 % (836 de 183978)
---> gk_handling: 0.4544 % (836 de 183978)
---> gk_kicking: 0.4544 % (836 de 183978)
---> gk_positioning: 0.4544 % (836 de 183978)
---> gk_reflexes: 0.4544 % (836 de 183978)

Analizando nulos de team.data
---> team_fifa_api_id: 3.6789 % (11 de 299)

Analizando nulos de teamAtt.data
---> buildUpPlayDribbling: 66.4609 % (969 de 1458)

Analizando nulos de ginf.data
---> odd_over: 90.3382 % (9135 de 10112)
---> odd_under: 90.3382 % (9135 de 10112)
---> odd_bts: 90.3382 % (9135 de 10112)
---> odd_bts_n: 90.3382 % (9135 de 10112)

Analizando nulos de events.data
---> event_type2: 77.2273 % (726716 de 941009)
---> player: 6.4543 % (60736 de 941009)
---> player2: 69.0425 % (649696 de 941009)
---> player_in: 94.5043 % (889294 de 941009)
---> player_out: 94.5019 % (889271 de 941009)
---> shot_place: 75.8282 % (713550 de 941009)
---> shot_outcome: 75.7178 % (712511 de 941009)
---> location: 50.3653 % (473942 de 941009)
---> bodypart: 75.6448 % (711824 de 941009)
---> situation: 75.6499 % (711872 de 941009)

Analizando nulos de estadios.data
```

Figura B.4: Resultado del estudio de valores nulos (2 de 2).

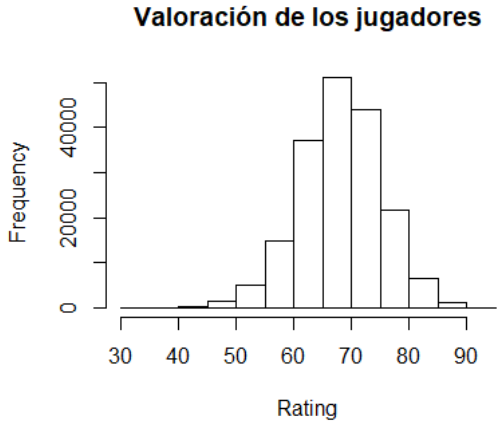

Figura B.5: Histograma de valoraciones.

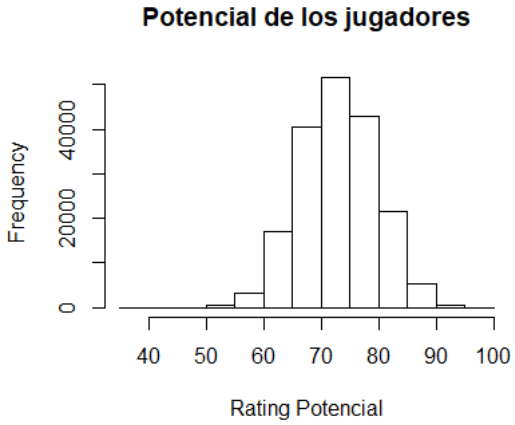

Figura B.6: Histograma de potenciales.

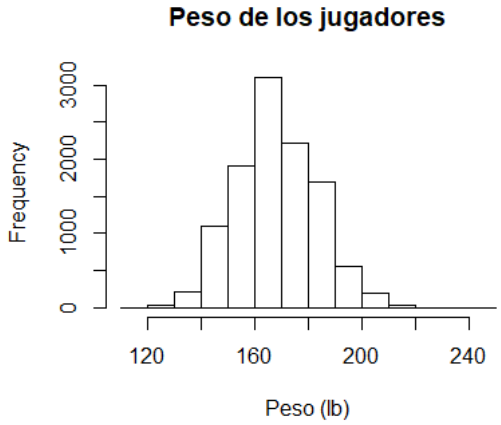

Figura B.7: Histograma de pesos.

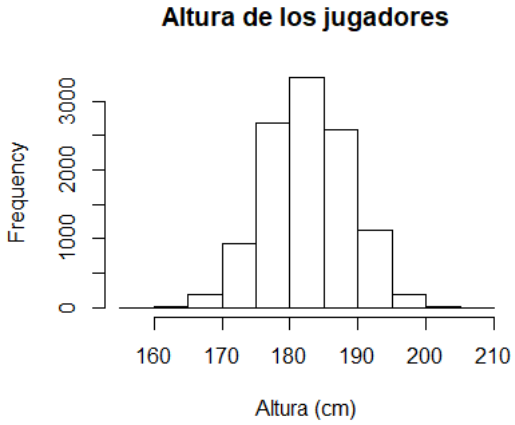

Figura B.8: Histograma de alturas.

B.2. Esquemas relacionales

En esta sección se muestran los esquemas relacionales de ambos conjuntos de datos iniciales, así como algunos comentarios sobre los mismos y sus datos. Estos esquemas se han realizado con la herramienta draw.io [66].

El esquema relacional completo del conjunto de datos de eventos se puede ver en la Figura B.9. En cuanto a los atributos interesantes de la tabla 'GINF', id_osp es el id del partido, adv_stats indica si se tienen eventos del mismo, date, league, season y country son suficientemente descriptivos, ht y at son los nombres del equipo de casa y el de fuera, respectivamente (*home* y *away*), fthg y ftag son el número de goles marcados por cada equipo, y los que comienzan por odd están relacionados con apuestas. En cuanto a los de la tabla 'EVENTS', los únicos atributos no numéricos son text (el comentario del evento), event_team, opponent, player, player2, player_in y player_out, todos ellos cadenas de caracteres. player y player2 son los jugadores involucrados en el evento, a menos que éste sea una sustitución, en cuyo caso utiliza player_in y player_out. Los enteros de la mayoría de los demás atributos son códigos, que se podían comprobar en un documento adjunto al conjunto de datos.

El conjunto de datos de alineaciones, cuyo esquema relacional reducido puede verse en la Figura B.10, está mucho más normalizado, utilizando IDs para los equipos y jugadores, a diferencia del anterior. En 'MATCH' interesan, además de los IDs generales, las coordenadas X e Y de los once jugadores titulares de cada equipo, los IDs de dichos jugadores (son los atributos home/away_player_X) y la información básica del partido. Las demás tablas son bastante descriptivas, aunque se decidió traducir todo al castellano para que todo el proyecto esté estandarizado en un mismo lenguaje. Merece la pena comentar que los atributos que terminan en 'Class' en la tabla 'TEAM_ATTRIBUTES' son textuales, con pocos valores posibles en cada uno, y que resumen los atributos de su mismo nombre, numéricos. También son numéricos (entre 0 y 100) todos los atributos de la tabla 'PLAYER_ATTRIBUTES', a excepción de date, preferred_foot, y los dos work_rates.

Nótese que, en este esquema, en la tabla 'MATCH' se ha reducido el número de atributos, tanto en los que sus nombres son iguales (enumerando del 1 al 11) como en atributos sobre cuotas de apuestas, como 365H o PSA. En el primer caso se han puesto puntos suspensivos para indicarlo y en el segundo, como los atributos no son de interés para este estudio y sus nombres eran poco explicativos, se han mencionado de pasada en el último atributo, para que el esquema fuera más corto y, por ende, más legible.

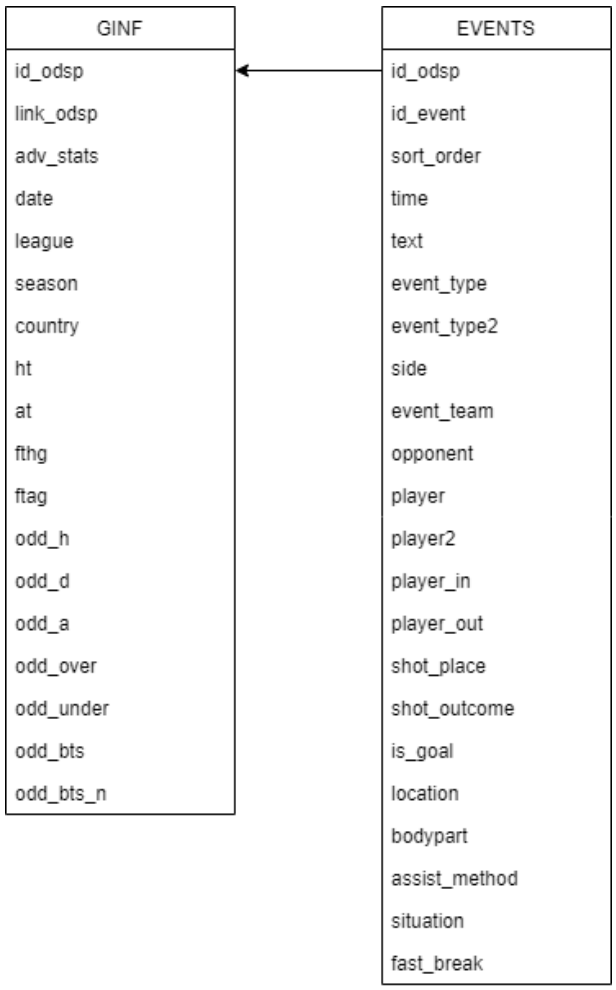

Figura B.9: Esquema del conjunto de datos de eventos.

También aparecen estadísticas generales, pero eran erróneas en muchos casos porque el autor del conjunto de datos recolectó erróneamente algunos datos, por lo que se han utilizado las de los eventos finalmente, ya que además su grano era menor.

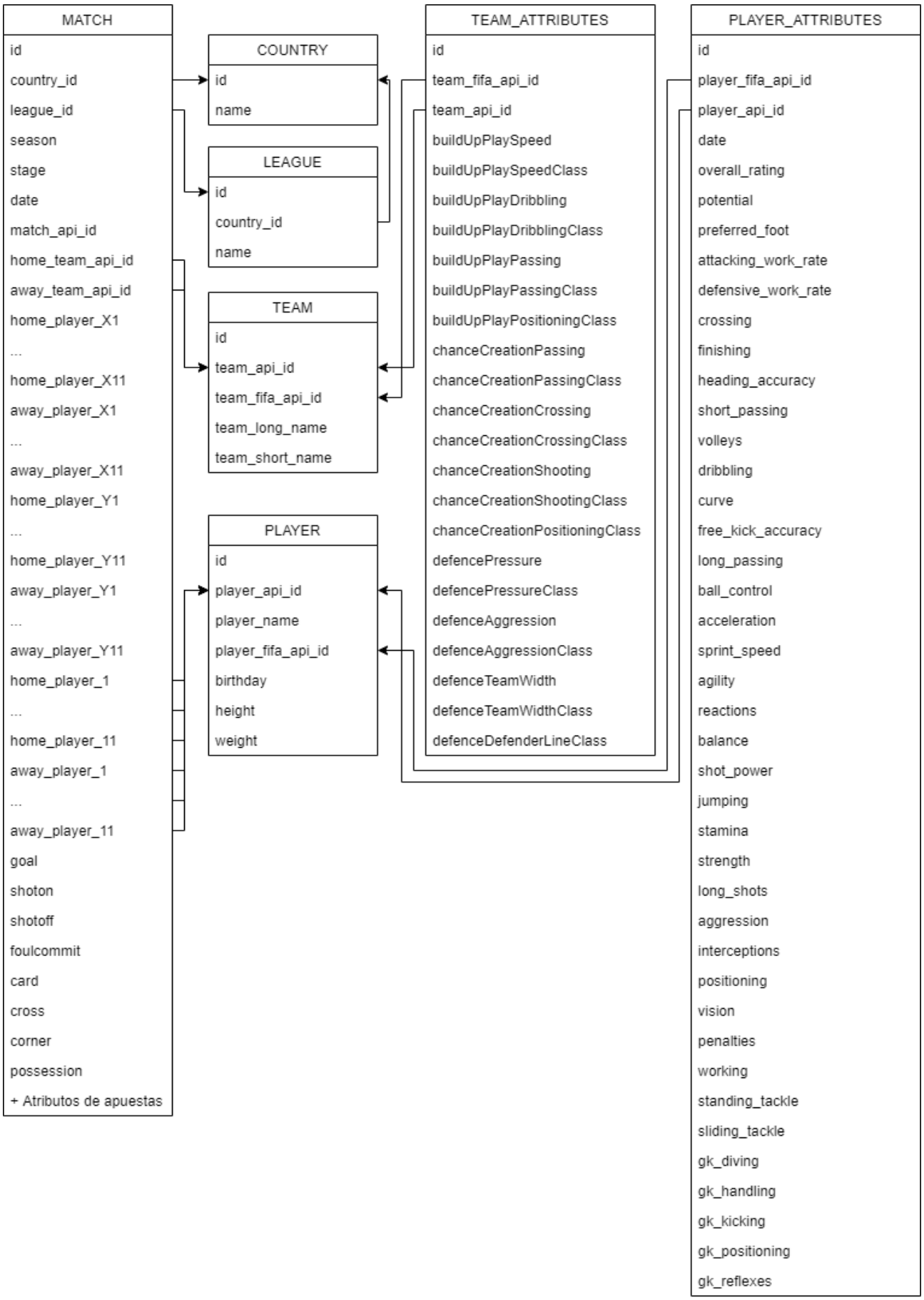

Figura B.10: Esquema del conjunto de datos de alineaciones.

B.3. Limitaciones

En esta sección se van a detallar las limitaciones de los conjuntos de datos que se han utilizado para el proyecto. Se pueden dividir en dos tipos de limitaciones: las de datos con los que se habría deseado contar pero de los que no se dispone, como los árbitros de cada partido, y las de datos que en algunas tuplas no existían cuando debían haberlo hecho, como la falta de eventos en un partido. Las de este segundo tipo vendrán acompañadas de la explicación de cómo se ha solucionado.

Estos son algunos de los tipos de datos interesantes de los que no se dispone:

- Posicionamiento dinámico de los jugadores en el campo. Solo se tienen las coordenadas del once inicial. Por ello, se ha supuesto que si un jugador sustituye a otro se coloca en la misma posición en el campo que el anterior.

- Lesiones y sanciones. La ausencia en ciertos partidos de algunos jugadores que podrían ser importantes para el equipo podría deberse a lesiones o sanciones (y no a decisiones técnicas del entrenador) pero no se dispone de ningún tipo de información sobre ellas. Además, sería interesante poder mostrarle al usuario final qué jugadores tiene disponibles su equipo y cuáles no debido a estas razones.

- Árbitros. Con el árbitro principal de cada partido se podría haber analizado el rendimiento de los equipos en función de los árbitros, además de poder analizar la tendencia de cada árbitro en cuanto a dejar seguir el juego o pararlo y sacar muchas tarjetas.

- Entrenadores. Además de clubes y jugadores, se podría haber analizado la carrera y el rendimiento de los entrenadores.

- Estadios de cada partido. Como los conjuntos de datos no contenían esta información, se ha inferido a partir del estadio del equipo local.

- Partidos de selecciones nacionales y/o de competiciones europeas. Podrían ser significativos si ocurren en medio de una temporada y hacen que un jugador se canse o incluso se lesione, además de poder provocar paradas en los calendarios de las ligas regulares.

- Información económica. Disponer de información económica tanto de los clubes (límite salarial o capital disponible para fichajes) como de jugadores (valor de mercado, años de contrato restantes o salario) habría aportado otro enfoque.

- Información histórica de equipos y jugadores. Saber los años en activo que llevan los jugadores para tener en cuenta su grado de experiencia o el palmarés de los equipos para conocer su influencia habría sido interesante.

- Posición natural de cada jugador. Al menos para la visualización de los cuadros de mando finales, habría convenido contar con la posición habitual o natural de cada jugador. No se ha inferido de las coordenadas donde juega cada uno por la misma razón que se comenta en la Sección 3.2.3, además de que hay jugadores muy polivalentes que no tienen una posición natural concreta.

- Puntos Fantasy de los jugadores. Si se hubiera dispuesto de información sobre los llamados *Puntos Fantasy* [67] de los jugadores (que resumen cómo de bien ha jugado cada jugador en cada jornada), se habría tenido más información de los mismos, además de la confianza de los usuarios de estas aplicaciones en los jugadores, en función de si los alinean para sus equipos virtuales o no. Esta confianza podría incluso ayudar a cuantificar lo que dice la prensa deportiva de los jugadores, ya que muchos usuarios los alinean en función de lo que leen sobre ellos.

- Más tipos de eventos. Sería significativo haber contado también con información como los pases, los intentos de pase, las paradas o los robos de balón para poder analizar mejor otras posiciones del campo, ya que actualmente lo más fácilmente analizable es el rendimiento ofensivo. Con toda esa información habría sido posible incluso crear un sistema de puntos como los *Puntos Fantasy* mencionados en el punto anterior, que habría ayudado a elegir las alineaciones o como otro punto interesante a desarrollar.

Por otra parte, estos son algunos casos concretos de los datos inexistentes o erróneos en los conjuntos de datos iniciales, así como las soluciones tomadas:

- Partidos sin alineaciones o eventos. Existen partidos para los que hay eventos pero no hay alineaciones y partidos para los que hay alineaciones pero no eventos. No se ha podido remediar ya que no se dispone de esos datos, por lo que se le ha indicado al usuario mediante el nombre de las tablas de hechos, añadiendo "Conocidos/as", para enfatizar que la información de dichas tablas puede no estar completa.

- Posiciones incorrectas. Algunos partidos tenían posiciones incorrectas, como que el portero no estuviera colocado en el sitio habitual. Estos casos se han arreglado manualmente, ya que eran pocos.

- Partidos de Premier League sin eventos. En relación con el primer punto de esta lista, ningún partido de la Premier League de las temporadas 2011/2012 y 2012/2013 cuenta con eventos. De esta forma, se mantuvo la información de Fact_AlineacionesConocidas (se supuso que no hubo ningún cambio) y en las tablas de hechos agregadas se colocaron indicadores en los atributos de los que no se disponía información, como los goles marcados por los clubes. Este indicador sirve para que los usuarios excluyan fácilmente estos partidos de las consultas de los goles y demás para no adulterar los resultados.

- Fallos en las sustituciones. En algunos eventos de sustituciones, los jugadores implicados estaban almacenados al revés (*player-in* debía ser *player-out* y viceversa). Para solucionarlo, se comprobaba en los procesos ETL si el jugador que salía del campo ya estaba jugando. En ese caso, el cambio era correcto. En caso de que el que estaba jugando fuera el que se suponía que iba a entrar, se le daba la vuelta al cambio. Si ninguno estaba jugando, se descartaba la sustitución.

- Jugadores involucrados en penaltis. En los eventos no se incluye qué jugador cometió el penalti ni qué jugador lo recibió. Sin embargo, en el comentario de cada evento sí aparecen los nombres de los que los han cometido. Estos aparecían, además, de dos formas distintas, que se pueden ver en la Figura B.11. Para solucionar ambos casos se utilizaron expresiones regulares con el nodo de KNIME "String Manipulation". En el primer caso, el nodo contenía la expresión *regexReplace($text$,"Penalty conceded by ([a-zA-Z-']+) .*", "$1")* y en el segundo, la expresión *regexReplace($text$,"[a-zA-Z0-9-'.]+[.]1 ([a-zA-Z-']+) draws.*", "$1")*.

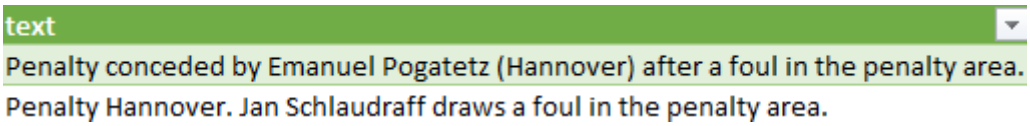

Figura B.11: Comentarios de penaltis en el conjunto de datos de eventos.

- Falta de estadios. En Wikidata no estaban todos los estadios necesarios (los de los partidos de las cinco temporadas en las cinco ligas). Como solo faltaban cuatro estadios, se añadió su información manualmente. Estos estadios eran el *SGL Arena*, el *Sardegna Arena*, el *Estadio Dino Manuzzi* y el *HDI-Arena*.

- Sin criterio de actualización de versiones. En el conjunto de datos de partidos no se sigue un criterio de periodicidad a la hora de recopilar nueva información de versiones de jugadores o partidos, que habría normalizado ligeramente la versión final del almacén de datos para su visualización.

- Errores en la codificación del fichero de eventos. En el fichero 'events.csv', correspondiente al conjunto de datos de eventos, había muchos nombres de jugadores mal codificados, de forma que aparecían más nombres diferentes que jugadores representaban. Se arregló de forma manual con expresiones regulares para evitar problemas en la posterior integración de los nombres con el otro conjunto de datos. Se pasó de tener 6352 nombres a tener 5781.

- Errores en nombres de jugadores. En el fichero 'player.csv', del conjunto de datos de alineaciones, hay nombres incorrectos ya que, tras el nombre, aparece un número (ejemplo: "Javier García, 18"). Se cambió manualmente, con una expresión regular.

- Mismas personas con diferentes nombres en el mismo conjunto de datos. En el conjunto de datos de eventos, en ocasiones un mismo jugador aparece con distintos nombres en diferentes eventos (por ejemplo, 'Ander Herrera' - 'Ander'). Se ha tratado de solucionar en los procesos ETL buscando la correspondencia de ambos (por separado) con el verdadero jugador en el otro conjunto de datos.

- Diferentes personas con el mismo nombre en el mismo conjunto de datos. Cabe la posibilidad de que esto haya ocurrido en el conjunto de eventos, aunque no es habitual. No ha habido manera de distinguir dos jugadores que tienen el mismo nombre, ya que en este conjunto el nombre es el identificador.

Hand ball by Stevan Jovetic (Inter Milan).	2	AC Milan	Internazionale
Hand ball by Luiz Adriano (Milan).	2	AC Milan	Internazionale
Hand ball by Duje Cop (MAÂ¡laga).	1	Malaga	Eibar
Hand ball by Marko Livaja (Empoli).	1	Empoli	Napoli
Hand ball by Pierre Bengtsson (1. FSV Mainz 05).	2	Mainz	Schalke 04
Hand ball by Zlatko Junuzovic (SV Werder Bremen).	2	Werder Bremen	TSG Hoffenheim
Hand ball by Jannik Vestergaard (SV Werder Bremen).	2	Werder Bremen	TSG Hoffenheim
Hand ball by Nicola Sansone (Sassuolo).	1	Sassuolo	Atalanta
Hand ball by Gregoire Defrel (Sassuolo).	1	Sassuolo	Atalanta
Hand ball by Sime Vrsaljko (Sassuolo).	1	Sassuolo	Atalanta
Hand ball by Gaetano Monachello (Atalanta).	2	Atalanta	Sassuolo

Figura B.12: Prueba de fallo en los datos (Faltas por mano).

- Errores en datos de los eventos. En ocasiones, los equipos están intercambiados en algún evento. Se puede observar el cambio de criterio en la Figura B.12: en los dos primeros eventos se puede ver que jugadores de distintos equipos han tocado el balón con la mano y en ambas ocasiones estas faltas han sido marcadas como provocadas por el AC Milan (primer equipo nombrado), mientras que en los dos últimos eventos de la imagen se puede ver cómo las mismas infracciones, cometidas por jugadores de distintos equipos, son marcadas para sus respectivos

equipos, siguiendo el criterio de la mayoría de eventos del fichero. Otro ejemplo se puede ver en la Figura B.13, donde dos fueras de juego del mismo equipo son marcados para dicho equipo en el primer caso y para el rival en el segundo.

Offside, Inter Milan. Mauro ZAÂ¡rate tries a	2	Internazionale	Parma
Offside, Inter Milan. Lucio tries a through b	2	AC Milan	Internazionale

Figura B.13: Prueba de fallo en los datos (Fueras de juego).

Anexo C

Detalles de diseño del almacén de datos

En este anexo se van a desarrollar algunos detalles relacionados con el diseño del almacén de datos. En el Anexo C.1 se muestra la matriz de bus del almacén de datos, mientras que en el Anexo C.2 se detallan los identificadores para situaciones especiales de sus diferentes dimensiones y en el Anexo C.3 se explica el significado de los atributos de la dimensión DIM_TipoEvento.

C.1. Matriz de bus del almacén de datos

En esta sección se incluye la matriz de bus del almacén de datos [68], que sirve para comprobar de una forma visual qué dimensiones afectan a qué *data marts*. Las filas representan los *data marts* (incluidos los de tablas de hechos agregadas) y las columnas, las dimensiones. Se han dividido en varias tablas (Tablas C.1 a C.5) por motivos de espacio.

C.2. IDs especiales en las dimensiones

Dado que faltan eventos y alineaciones en algunos partidos, y otra información en otros casos y, de acuerdo a la metodología de Kimball [43], no se desea almacenar ningún tipo de valores nulos (ni en las tablas de hechos [69] ni en las dimensiones

	DIM_Fecha	DIM_Liga	DIM_Estadio
Fact_EventosConocidos	X	X	X
Fact_AlineacionesConocidas	X	X	X
AggFact_JugadorPartido	X	X	X
AggFact_Partido	X	X	X
AggFact_EquipoTemporada		X	

Tabla C.1: Matriz de bus de almacén de datos (1 de 5).

	DIM_Jugador	DIM_VersionJugador	DIM_DatosJugador
Fact_EventosConocidos	X	X	X
Fact_AlineacionesConocidas	X	X	X
AggFact_JugadorPartido	X	X	X
AggFact_Partido			
AggFact_EquipoTemporada	X	X	X

Tabla C.2: Matriz de bus de almacén de datos (2 de 5).

	DIM_Equipo	DIM_VersionEquipo	DIM_TipoEvento
Fact_EventosConocidos	X	X	X
Fact_AlineacionesConocidas	X	X	
AggFact_JugadorPartido	X	X	
AggFact_Partido	X	X	
AggFact_EquipoTemporada	X	X	

Tabla C.3: Matriz de bus de almacén de datos (3 de 5).

	DIM_DetallesEvento	temporada	jornada	minuto
Fact_EventosConocidos	X	X	X	X
Fact_AlineacionesConocidas		X	X	
AggFact_JugadorPartido		X	X	
AggFact_Partido		X	X	
AggFact_EquipoTemporada		X		

Tabla C.4: Matriz de bus de almacén de datos (4 de 5).

	formacionCasa	formacionFuera	formacionMasUtilizada
Fact_EventosConocidos			
Fact_AlineacionesConocidas	X	X	
AggFact_JugadorPartido			
AggFact_Partido	X	X	
AggFact_EquipoTemporada			X

Tabla C.5: Matriz de bus de almacén de datos (5 de 5).

[70]), se han añadido algunas tuplas con valores especiales que indican que algo es desconocido o inexistente. En esta sección se va a detallar en qué dimensiones existen estos valores y cuáles son sus significados:

- idJugador: Vale 0 cuando no se sabe qué jugador ha realizado un evento o ha jugado (desconocido), y vale -1 cuando es un jugador inexistente. Este último caso se utiliza cuando en un partido no se utilizan los tres cambios, ya que en Fact_AlineacionesConocidas, por ejemplo, siempre se almacenan catorce elementos en cada uno de los arrays, por lo que si el último cambio no se ha utilizado se añadirá un -1 en el último elemento (que apunta a una tupla especial).

Ocurre algo parecido en AggFact_EquipoTemporada, en el caso de que la plantilla cuente con menos de 40 jugadores.

- idDatosJugador: Cuando no se dispone de datos físicos del jugador vale 0.
- idVersionJugador: Cuando vale 0 apunta al jugador desconocido, de id 0. Cuando vale -1, al jugador inexistente, de id -1.
- idDetallesEvento: Cuando no se dispone de los detalles del evento, este atributo vale 0, así como los demás atributos de su dimensión.
- AggFact_Partido: Que las métricas valgan -1 indica que no se tienen eventos de ese partido, por lo que solo se ha almacenado ese partido ya que se disponía de las alineaciones e información básica del mismo.

C.3. Atributos de DIM_TipoEvento

En esta sección se va a detallar qué indica cada uno de los atributos de la dimensión DIM_TipoEvento, en relación con la tabla de hechos Fact_EventosConocidos, la única del almacén de datos que la utiliza. Sirve a modo de manual o documentación para el usuario.

Hay un par de casos que conviene destacar: golMarcado indica si el jugador del evento ha metido un gol "correcto" (al rival), mientras que golEnPropia indica si ese jugador se lo ha metido en propia portería. Otra alternativa, descartada, es que hubiera un atributo que indicara si ha habido gol y otro que especificara si ha ocurrido en propia meta. Por otra parte, tarjetaRoja indica si ha habido una expulsión, segundaTarjetaAmarilla indica si ha habido una segunda tarjeta amarilla (que indica expulsión) y tarjeta amarilla indica si ha habido tarjeta amarilla. Por ello, si se muestra una segunda tarjeta amarilla los tres atributos valdrán 1. Por último, que asistencia valga 1 indica que el gol ocurrido en el evento ha derivado de una asistencia, cuyo autor es el jugador2 indicado en la tabla de hechos.

A continuación se detallan todos los atributos:

- golMarcado: Vale 1 si idJugador1 ha metido gol (idJugador1 pertenece a idEquipoEvento, y le ha metido gol al otro equipo).
- golEnPropia: Vale 1 si idJugador1 (que pertenece a idEquipoEvento), se ha metido un gol en propia puerta. Por ello, el gol subiría al marcador del equipo que no es el del evento.

- asistencia: Vale 1 si idJugador2 le ha dado una asistencia de gol a idJugador1 y éste ha metido gol (consecuentemente, solo se da en casos en los que golMarcado vale 1, aunque no todos los goles tienen asistencias).
- tiroAPuerta: Vale 1 si idJugador1 ha disparado a puerta.
- tiro: Vale 1 si idJugador1 ha realizado un tiro.
- tiroAlPoste: Vale 1 si idJugador1 ha tirado a uno de los dos postes o al larguero.
- tarjetaRoja: Vale 1 si idJugador1 ha sido expulsado (tanto con roja directa como con doble amarilla).
- tarjetaAmarilla: Vale 1 si idJugador1 ha recibido una tarjeta amarilla (tanto por primera vez como por segunda, en la que es expulsado).
- segundaTarjetaAmarilla: Vale 1 si idJugador1 ha recibido la segunda tarjeta amarilla (siempre que vale 1, tarjetaRoja y tarjetaAmarilla también valen 1).
- faltaCometida: Vale 1 si idJugador1 ha cometido una falta.
- libreDirectoGanado: Vale 1 si idJugador1 ha conseguido un libre directo.
- manoCometida: Vale 1 si idJugador1 ha cometido una mano (faltaCometida también vale 1).
- cornerConcedido: Vale 1 si idJugador1 ha concedido un corner al equipo contrario.
- fueraDeJuegoConcedido: Vale 1 si idJugador1 estaba en fuera de juego.
- penaltyConcedido: Vale 1 si idJugador1 ha cometido penalty (faltaCometida también vale 1).

Que cualquiera de los atributos valga 0 indica que dicho atributo no ha ocurrido.

Anexo D

Consultas de explotación

En este anexo se van a explicar las consultas analíticas que se han creado para explotar el almacén de datos sin necesidad de más artificios, así como los resultados de las mismas. Estas consultas tienen diferentes enfoques: el primero, en el Anexo D.1, es el del entrenador de un equipo (se busca encontrar datos que respalden las decisiones que toma o encontrar nuevas visiones que amplíen su perspectiva) y el segundo, en el Anexo D.2, trata de obtener estadísticas y datos con variadas finalidades. El código fuente de todas estas consultas, además de otras menos relevantes no incluidas en este anexo, puede encontrarse en el repositorio de GitHub del proyecto [53].

D.1. Consultas de un entrenador

Las consultas que podría realizar un entrenador buscan optimizar el rendimiento del equipo en su conjunto, encontrando puntos débiles para repararlos o puntos fuertes para explotarlos, con el fin último de aumentar sus probabilidades de ganar partidos. Para mostrarlo, se van a plantear una serie de situaciones no ficticias.

El FC Barcelona afronta un partido contra uno de los mejores equipos de la liga española: el Sevilla FC. Su entrenador quiere saber a qué jugador le motiva más jugar contra este equipo, en función de la información ofensiva de partidos anteriores contra el propio Sevilla. Para ello, se han realizado varias consultas similares: los cinco jugadores del FC Barcelona que más goles le han metido al Sevilla FC, los cinco que más asistencias han dado, o combinaciones de ambas. Finalmente, se ha realizado una consulta en la que se asignan dos puntos por gol marcado y 1,5 puntos por asistencia dada, de forma que se pondera la importancia de cada atributo. El resultado puede comprobarse en la Figura D.1.

Por otra parte, en lugar de mostrar los resultados absolutos, se ha realizado otra

Nombre Jugador	Partidos Jugados	Minutos Jugados	Goles Marcados	Asistencias
Lionel Messi	4	360	3	3
Neymar	4	344	2	2
Cesc Fabregas	4	176	3	0
Alexis Sanchez	4	215	2	1
David Villa	3	180	2	0

Figura D.1: Resultado consulta FC Barcelona - Sevilla FC (Mejor puntuación).

consulta (Figura D.2) con los jugadores que necesitan menos minutos para meter un gol o dar una asistencia.

Nombre Jugador	Partidos Jugados	Minutos Jugados	Goles Marcados	Asistencias	Minutos para GoA
Cesc Fabregas	4	176	3	0	58.6666667
Lionel Messi	4	360	3	3	60
Alexis Sanchez	4	215	2	1	71.6666667
Neymar	4	344	2	2	86
David Villa	3	180	2	0	90

Figura D.2: Resultado consulta FC Barcelona - Sevilla FC (Mejor tasa de minutos por gol o asistencia).

Es 11 de abril de 2013. El Real Zaragoza ha de jugar dentro de tres días contra el FC Barcelona. El entrenador del Real Zaragoza quiere saber en qué rango de minutos su equipo ha recibido más goles esta temporada, para aumentar la concentración en defensa en ese periodo de tiempo (Figura D.3).

0'-15'	16'-30'	31'-45'	46'-60'	61'-75'	76'-90'
6	10	9	7	7	6

Figura D.3: Resultado consulta Real Zaragoza - FC Barcelona.

13 días más tarde, el Real Zaragoza todavía no ha ganado un solo partido de liga en lo que va de año, y recibe al RCD Mallorca, un rival directo, en tres días. Como existe la necesidad de ganar, se va a lanzar al ataque. El entrenador desea saber en qué zona se le puede hacer más daño al rival, sabiendo dónde ha recibido más goles esta temporada (Figura D.4).

Las dos últimas consultas no tienen una situación ni un equipo concreto que se quiera beneficiar de ellas. Ambas tratan de encontrar equipos que deberían haber reconsiderado sus formaciones. La primera de ellas, cuyo resultado se puede ver en la Figura D.5, busca equipos que utilizan habitualmente un tridente ofensivo, esto es,

Localización	Goles Recibidos
Centro del área grande	32
Fuera del área	7
Muy cerca	7

Figura D.4: Resultado consulta Real Zaragoza - RCD Mallorca.

tres delanteros jugando simultáneamente (habitualmente un delantero centro y dos extremos) que, sin embargo, meten pocos goles (menos de un gol de media por partido).

Temporada	Equipo	Liga	Partidos	Goles A Favor	Goles En Contra	Formacion habit.
2015/2016	Aston Villa F.C.	Premier League	38	27	76	4-3-3
	Frosinone Calcio	Serie A	37	35	72	4-3-3
	Hellas Verona F.C.	Serie A	37	32	60	4-3-3
	Bologna F.C. 1909	Serie A	37	33	45	4-3-3
	FC Ingolstadt 04	1. Bundesliga	33	31	39	4-3-3
2014/2015	R.C. Lens	Ligue 1	38	32	61	4-3-3
	Aston Villa F.C.	Premier League	38	31	57	4-3-3
	Stade Rennais F.C.	Ligue 1	38	35	42	4-3-3
2013/2014	Calcio Catania	Serie A	38	34	66	4-3-3
	Granada CF	LIGA BBVA	38	32	56	4-3-3
	Aston Villa F.C.	Premier League	32	30	53	4-3-3

Figura D.5: Resultado consulta de formación ofensiva.

La segunda (Figura D.6) busca equipos que alinean habitualmente cinco defensas y, a pesar de ello, recibe más goles de los que debería (más de 1.2 goles de media por partido).

Temporada	Equipo	Liga	Partidos	Goles A Favor	Goles En Contra	Formacion habit.
2012/2013	A.C. ChievoVerona	Serie A	38	37	52	5-3-2

Figura D.6: Resultado consulta formación defensiva.

D.2. Otras consultas

Este segundo enfoque está menos estructurado, ya que las consultas no tienen un objetivo común. Se buscan desde curiosidades que le podrían interesar a un lector habitual de la prensa deportiva hasta algunas clasificaciones de trofeos individuales a nivel internacional, pasando por datos objetivos que podrían ayudar a un director deportivo a rastrear el mercado en busca de los jugadores que más se adapten a las necesidades del equipo.

En esta primera clasificación, que se puede ver en la Figura D.7, se van a mostrar los tres jugadores con más goles de cada temporada, de forma semejante al premio llamado 'Bota de oro'.

Temporada	Jugador	Goles	Equipo	Liga
2015/2016	Luis Suarez	37	FC Barcelona	LIGA BBVA
	Zlatan Ibrahimovic	36	Paris Saint-Germain	Ligue 1
	Gonzalo Higuain	33	Napoli	Serie A
2014/2015	Cristiano Ronaldo	44	Real Madrid C.F.	LIGA BBVA
	Lionel Messi	40	FC Barcelona	LIGA BBVA
	Alexandre Lacazette	27	Olympique Lyonnais	Ligue 1
2013/2014	Cristiano Ronaldo	31	Real Madrid C.F.	LIGA BBVA
	Luis Suarez	28	Liverpool F.C.	Premier League
	Lionel Messi	28	FC Barcelona	LIGA BBVA
2012/2013	Lionel Messi	46	FC Barcelona	LIGA BBVA
	Cristiano Ronaldo	34	Real Madrid C.F.	LIGA BBVA
	Zlatan Ibrahimovic	30	Paris Saint-Germain	Ligue 1
2011/2012	Lionel Messi	50	FC Barcelona	LIGA BBVA
	Cristiano Ronaldo	43	Real Madrid C.F.	LIGA BBVA
	Zlatan Ibrahimovic	28	A.C. Milan	Serie A

Figura D.7: Resultado consulta bota de oro.

Para la siguiente consulta, se han ampliado los conocimientos del lenguaje de programación PL/SQL [55], ya que solo se tenía una muy pequeña base del mismo. Realizar la consulta con dicho lenguaje ha sido imprescindible por la necesidad de manejar arrays. La consulta consiste en saber qué tres jugadores han marcado más goles saliendo como suplentes (Figura D.8).

```
MAXIMOS GOLEADORES SUPLENTES
============================
Mevlut Erdinc: 17 goles (en 67 partidos).
Alvaro Morata: 14 goles (en 48 partidos).
Cesc Fabregas: 12 goles (en 43 partidos).
```

Figura D.8: Resultado consulta goles de suplentes.

En la Figura D.9 se puede descubrir qué diez jugadores tienen rendimientos ofensivos más diferentes en función de si juegan en casa o fuera.

Jugador	Equipo	Diff	Gol. C	Asist. C	Gol. F	Asist. F	Part. C	Part. F
Antonio Di Natale	Udinese Calcio	50	58	16	18	6	82	65
Lionel Messi	FC Barcelona	48	113	42	77	30	80	82
Karim Benzema	Real Madrid C.F.	47	59	25	21	16	76	68
Cristiano Ronaldo	Real Madrid C.F.	40	108	30	77	21	84	79
Alexis Sanchez	FC Barcelona	30	29	16	7	8	43	40
Wissam Ben Yedder	Toulouse F.C.	30	44	11	18	7	76	71
Neymar	FC Barcelona	29	35	18	18	6	45	42
Fabrizio Miccoli	Unione Sportiva Citta di Palermo	28	21	15	3	5	33	24
Arjen Robben	FC Bayern Munich	25	31	16	14	8	54	44
Zlatan Ibrahimovic	Paris Saint-Germain	25	65	22	46	16	61	58

Figura D.9: Resultado consulta diferencia de jugadores entre casa y fuera.

Otra consulta interesante, que se puede ver en la Figura D.10, es saber cuáles son las diez parejas de jugadores que "se entienden mejor", es decir, que tienen más combinaciones de asistencia de uno y gol del otro.

Goles Totales	Jugador 1	Jugador 2	Goles J1	Goles J2
27	Cristiano Ronaldo	Karim Benzema	14	13
26	Luis Suarez	Lionel Messi	14	12
21	Lionel Messi	Pedro Rodriguez	12	9
20	Neymar	Lionel Messi	11	9
19	Cristiano Ronaldo	Gareth Bale	15	4
17	Lionel Messi	Cesc Fabregas	11	6
17	Lionel Messi	Alexis Sanchez	10	7
16	Cristiano Ronaldo	Angel Di Maria	14	2
16	Neymar	Luis Suarez	10	6
15	Thomas Mueller	Franck Ribery	9	6

Figura D.10: Resultado consulta parejas de jugadores que mejor se entienden.

En la consulta de la Figura D.11 se ha analizado cuáles son los estadios en los que más goles tardíos (del minuto 85 en adelante) ocurren.

Estadio	En Total	A Favor	En Contra
Stadio Olimpico	64	43	21
San Siro	51	26	25
Stadio Artemio Franchi	46	18	28
Stadio Luigi Ferraris	46	23	23
Stadio Marc'Antonio Bentegodi	41	22	19
Stadium Municipal	39	20	19
Parc des Princes	38	31	7
Estadio Santiago Bernabeu	37	31	6
Anoeta Stadium	36	20	16
Stade de la Mosson	36	25	11

Figura D.11: Resultado consulta estadios con goles tardíos.

Es el caso de un director deportivo que quiere fichar a un jugador sub-21 (futura promesa), que ya haya demostrado tener el potencial y la perseverancia necesarios para haberse dado a conocer en una de las grandes ligas gracias a sus goles y sus asistencias. La fecha actual es 1 de julio de 2015 y le interesa saber su edad actual. El resultado de la consulta puede comprobarse en la Figura D.12.

Jugador	Edad	Goles	Asistencias	Minutos	Equipo	Liga	Temporada
Domenico Berardi	20	14	6	1526	U.S. Sassuolo Calcio	Serie A	2013/2014
Domenico Berardi	20	13	4	1776	U.S. Sassuolo Calcio	Serie A	2014/2015
Nabil Fekir	21	9	7	1809	Olympique Lyonnais	Ligue 1	2014/2015
Paulo Dybala	21	9	7	1677	Unione Sportiva Citta di Palermo	Serie A	2014/2015
Harry Kane	21	12	3	1742	Tottenham Hotspur F.C.	Premier League	2014/2015
Julian Draxler	21	10	1	1501	FC Schalke 04	1. Bundesliga	2012/2013
Saido Berahino	21	10	1	1696	West Bromwich Albion F.C.	Premier League	2014/2015
Raheem Sterling	20	8	3	1409	Liverpool F.C.	Premier League	2013/2014
Anthony Martial	19	7	3	1278	AS Monaco FC	Ligue 1	2014/2015
Divock Origi	20	7	3	1410	Lille O.S.C.	Ligue 1	2014/2015
Hakan Calhanoglu	21	6	4	1426	Bayer 04 Leverkusen	1. Bundesliga	2014/2015
Raheem Sterling	20	3	7	1822	Liverpool F.C.	Premier League	2014/2015

Figura D.12: Resultado consulta mejores jugadores sub-21.

Y en relación con la anterior consulta, ¿con qué edad se obtiene habitualmente un mayor rendimiento ofensivo? (Figura D.13).

Edad	Goles
25	2130
26	2105
27	2096
24	2004
28	1820

Figura D.13: Resultado consulta edad con mayor rendimiento ofensivo.

Además, se ha querido realizar alguna consulta con operadores de agregación para ampliar conocimientos. Se ha utilizado el operador GROUPING SETS (que permite especificar exactamente qué agrupamientos a considerar para calcular subtotales) [56], para calcular el número de goles totales, por temporadas, por temporadas en las distintas ligas y por jornada (considerando una jornada como una ocurrida en una liga y temporada concreta). El resultado de esta consulta puede verse en la Figura D.14.

Por último, se han realizado varias consultas en busca de anomalías que podrían descubrir fraudes en algunos partidos, en relación con casas de apuestas. Estas consultas tan solo muestran datos extraños, que darían lugar posteriormente a un estudio más concienzudo de cada caso, con más datos involucrados en el mismo.

JORNADA	TEMPORADA	LIGA	GOLES	GOLESPORPARTIDO
			21765	2.6880326
	2013/2014		4767	2.76187717
	2015/2016		4747	2.65938375
	2014/2015		4650	2.62563523
	2012/2013		3931	2.73746518
	2011/2012		3670	2.6613488
	2012/2013	LIGA BBVA	1085	2.86279683
	2013/2014	LIGA BBVA	1045	2.75
	2013/2014	Serie A	1031	2.72031662
	2015/2016	Premier League	1031	2.72031662
	2014/2015	Serie A	1024	2.70184697
	2015/2016	LIGA BBVA	1010	2.72972973
	2012/2013	Serie A	1000	2.63852243
	2011/2012	LIGA BBVA	985	2.77464789
	2014/2015	Premier League	975	2.56578947
	2012/2013	Ligue 1	952	2.55227882
	2014/2015	Ligue 1	945	2.48684211
	2015/2016	Serie A	942	2.54594595
	2013/2014	Ligue 1	929	2.45767196
	2015/2016	Ligue 1	927	2.51219512
	2011/2012	Ligue 1	925	2.51358696
	2011/2012	Serie A	924	2.55248619
	2013/2014	Premier League	916	2.8625
	2014/2015	LIGA BBVA	906	2.64912281
	2012/2013	1. Bundesliga	894	2.93114754
	2013/2014	1. Bundesliga	846	3.14498141
	2015/2016	1. Bundesliga	837	2.81818182
	2011/2012	1. Bundesliga	836	2.84353741
	2014/2015	1. Bundesliga	800	2.75862069
38	2014/2015	Serie A	47	4.7
35	2014/2015	Serie A	42	4.2
4	2014/2015	LIGA BBVA	42	4.2
2	2013/2014	Serie A	42	4.2
31	2013/2014	Premier League	42	4.2
14	2013/2014	LIGA BBVA	42	4.2
20	2015/2016	LIGA BBVA	41	4.1
38	2014/2015	LIGA BBVA	41	4.1
7	2015/2016	Premier League	41	4.1
8	2014/2015	Premier League	40	4
15	2012/2013	Serie A	39	3.9
[...]				
25	2014/2015	Serie A	16	1.6
32	2011/2012	1. Bundesliga	15	1.66666667
19	2014/2015	1. Bundesliga	15	1.66666667
32	2014/2015	Ligue 1	15	1.5
3	2013/2014	Ligue 1	15	1.5
3	2014/2015	Ligue 1	14	1.4
16	2015/2016	Ligue 1	13	1.3
1	2015/2016	LIGA BBVA	12	1.2
23	2014/2015	Ligue 1	10	1

Figura D.14: Resultado consulta número de goles.

Para empezar, en la Figura D.15 se van a mostrar los partidos con varios goles en propia puerta.

Otro campo muy tratado en las apuestas, aunque no lo parezca, es el de los saques

Fecha	Casa	Res. C.	Res. Fu.	Fuera	GP C.	GP F.	GP Tot.
30/04/2015	Empoli F.C.	4	2	Napoli	1	2	3
07/12/2013	Liverpool F.C.	4	1	West Ham United F.C.	1	2	3
13/01/2016	Stoke City F.C.	4	1	Norwich City F.C.	0	2	2
13/01/2016	Chelsea F.C.	3	2	West Bromwich Albion F.C.	0	2	2
13/12/2014	F.C. Nantes	2	1	F.C. Girondins de Bordeaux	1	1	2
16/01/2016	Atalanta B.C.	1	1	FC Inter Milan	1	1	2
16/12/2012	A.C. Milan	4	1	Delfino Pescara 1936	0	2	2
18/10/2014	Southampton F.C.	8	0	Sunderland A.F.C.	0	2	2
19/10/2014	Queens Park Rangers F.C.	2	3	Liverpool F.C.	2	0	2
22/11/2014	AS Monaco FC	2	2	Stade Malherbe Caen	1	1	2
26/04/2014	Southampton F.C.	2	0	Everton F.C.	0	2	2
27/04/2014	Villarreal Club de Futbol	2	3	FC Barcelona	2	0	2
08/08/2015	Chelsea F.C.	4	4	Swansea City A.F.C.	0	2	2
01/02/2015	A.C. ChievoVerona	1	2	Napoli	1	1	2
05/04/2014	Paris Saint-Germain	3	0	Stade de Reims	0	2	2

Figura D.15: Resultado consulta varios goles en propia puerta.

de esquina. Por ello, la siguiente consulta (Figura D.16) muestra partidos con una gran cantidad de estos. Cabe recalcar que, con otra consulta, se ha comprobado que la media de saques de esquina por partido es 8.7.

Fecha	Casa	Fuera	Liga	Corners
25/11/2013	A.S. Roma	Cagliari	Serie A	26
03/10/2015	Norwich City F.C.	Leicester City F.C.	Premier League	25
05/10/2013	Valenciennes F.C.	Stade de Reims	Ligue 1	25
28/12/2015	West Bromwich Albion F.C.	Newcastle United F.C.	Premier League	24
13/01/2016	Tottenham Hotspur F.C.	Leicester City F.C.	Premier League	23
11/02/2012	Real Betis Balompie	Athletic Club	LIGA BBVA	23
11/03/2012	Genoa Cricket and Football Club	Juventus F.C.	Serie A	23
04/05/2013	1. FC Nurnberg	Bayer 04 Leverkusen	1. Bundesliga	22
23/08/2015	Lille O.S.C.	F.C. Girondins de Bordeaux	Ligue 1	22
16/03/2013	Bayer 04 Leverkusen	FC Bayern Munich	1. Bundesliga	22

Figura D.16: Resultado consulta muchos saques de esquina.

También se han realizado otras consultas similares, como goles en propia puerta más tempraneros (anteriores al minuto 6), grandes diferencias entre faltas cometidas y tarjetas recibidas o grandes diferencias entre el número de corners señalados entre ambas partes del partido, que se pueden comprobar, como se ha señalado, en el repositorio de GitHub [53].

Anexo E

Cuadros de mandos realizados y su creación

En este anexo se van a mostrar y explicar los *dashboards* creados en Microsoft PowerBI [34] que sirven como informes para entrenadores, así como sus procesos de creación en la herramienta.

Para empezar, se importaron los datos y se indicaron las relaciones entre las tablas. Cabe destacar que el programa tan solo permite una relación activa por tabla, lo que impidió posteriormente el uso completamente correcto de las mismas y obligó a crear tablas auxiliares para mostrar la información convenientemente.

Se han creado cuatro cuadros de mandos. En el Anexo E.1 se detalla el cuadro de rendimiento de equipos, en el Anexo E.2 el de estadísticas de partidos, en el Anexo E.3 el cuadro de alineaciones de partidos, y en el Anexo E.4, el de aptitudes de jugadores. Estos cuadros de mandos son solo una muestra del potencial que tiene el almacén de datos. Otras opciones a poder realizar pasan por observar la evolución de las aptitudes de los jugadores o crear infogramas de partidos con partidos previos jugados entre los mismos equipos (los llamados *Head-to-Head* o H2H) y sus rachas actuales.

E.1. Rendimiento de equipos

El primer cuadro, correspondiente a la Figura E.1, es el titulado "Rendimiento de equipos". En él, el entrenador ha de seleccionar su equipo (en el caso de que el informe estuviera hecho para un equipo concreto, se podría seleccionar dicho equipo en un filtro y no darle esa opción al entrenador, aunque de esta forma puede elegir también los equipos rivales para ver sus jugadores más peligrosos) y, opcionalmente, seleccionar una temporada y/o un rango de fechas. De esta forma, se le muestra el nombre de su equipo y estadísticas de juego en forma de tabla de una sola fila durante

el periodo seleccionado. Debajo aparecen dos gráficos de tablas: uno muestra los cinco jugadores que más minutos han jugado con dicho equipo en dicho periodo, y otro los cinco jugadores que más goles han metido en ese mismo periodo y equipo.

Para la última gráfica mencionada se ha utilizado el objeto visual "Infographic Designer" [72] del *marketplace* de PowerBI. Éste se ha usado, especialmente, para hacer el gráfico más visual: cada gol marcado por el jugador mencionado en el eje X es representado por un balón de fútbol, además de que la altura máxima de la columna indica el número de goles totales en el eje Y, sin necesidad de contar todos los balones. Para conseguir esto hay que, tras elegir la información como un gráfico de columnas común, seleccionar el 'Mark Designer' que aparece, y ahí seleccionar el icono a utilizar, darle la forma deseada, habilitar la opción de múltiples unidades y seleccionar que cada unidad es un gol y que cada fila tenga dos balones. Se podría haber realizado algo similar para la gráfica de minutos jugados, indicando que cada pequeño reloj que apareciera fueran, por ejemplo, 200 minutos jugados. Se ha desechado dado que no sería entendible a simple vista y no sería tan preciso como éste.

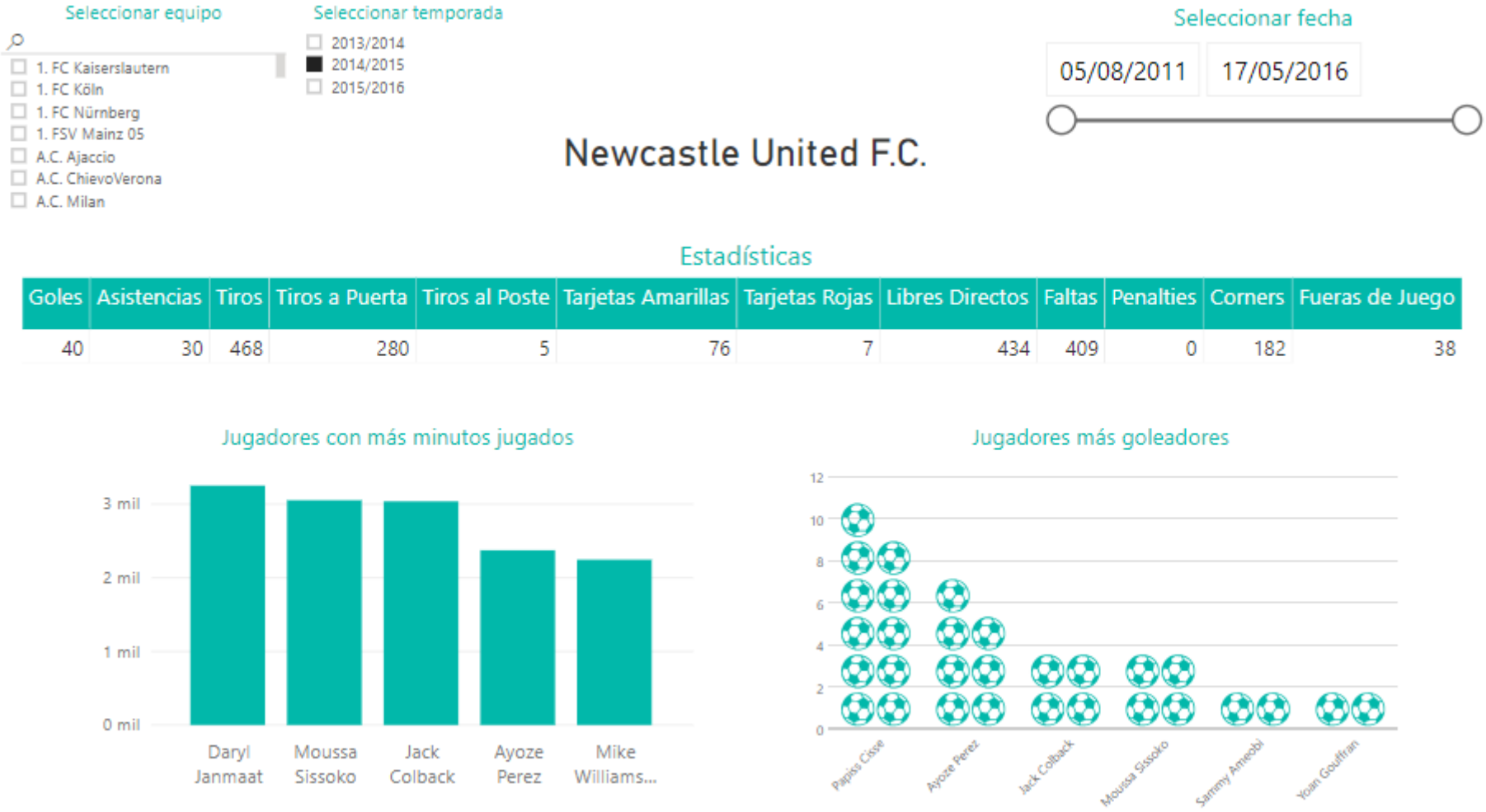

Figura E.1: Cuadro de rendimiento de los equipos.

E.2. Estadísticas de partidos

El segundo cuadro, correspondiente a la Figura E.2, titulado "Estadísticas de partidos", muestra las estadísticas del partido seleccionado (es necesario seleccionar equipo de casa, de fuera y fecha, todo de forma dinámica conforme se van eligiendo). Además de la información típica del partido (jornada, temporada, estadio, equipos, resultado y fecha), mostrados con texto simple, aparecen las estadísticas del partido como las faltas o los fueras de juego.

Para el gráfico principal se ha utilizado un peculiar pero acertado objeto visual llamado "Tornado" [73], también obtenido del *marketplace*. Para que muestre los datos de la forma deseada, requiere un atributo que indique la estadística que se está tratando, y un atributo por cada dato (casa y fuera), por lo que hay una tupla por cada estadística que se desea representar y partido. Para lograrlo, se ha vuelto a utilizar KNIME [32] (ante la lentitud de PowerQuery en operaciones de *unpivoting* con tantos datos, necesarias para conseguir una tupla por cada estadística).

En KNIME, primero se ha realizado una serie de operaciones *join* para tener toda la información deseada sobre los partidos, partiendo de AggFact_Partido. Después, se han hecho dos *unpivoting* (casa y fuera) para tener el grano de las tuplas deseado y se han filtrado las tuplas en función de si coincidían las estadísticas que se estaban tratando en casa y fuera. Así, se ha creado un fichero CSV llamado 'Info_Partido', que ha sido el utilizado finalmente en el gráfico.

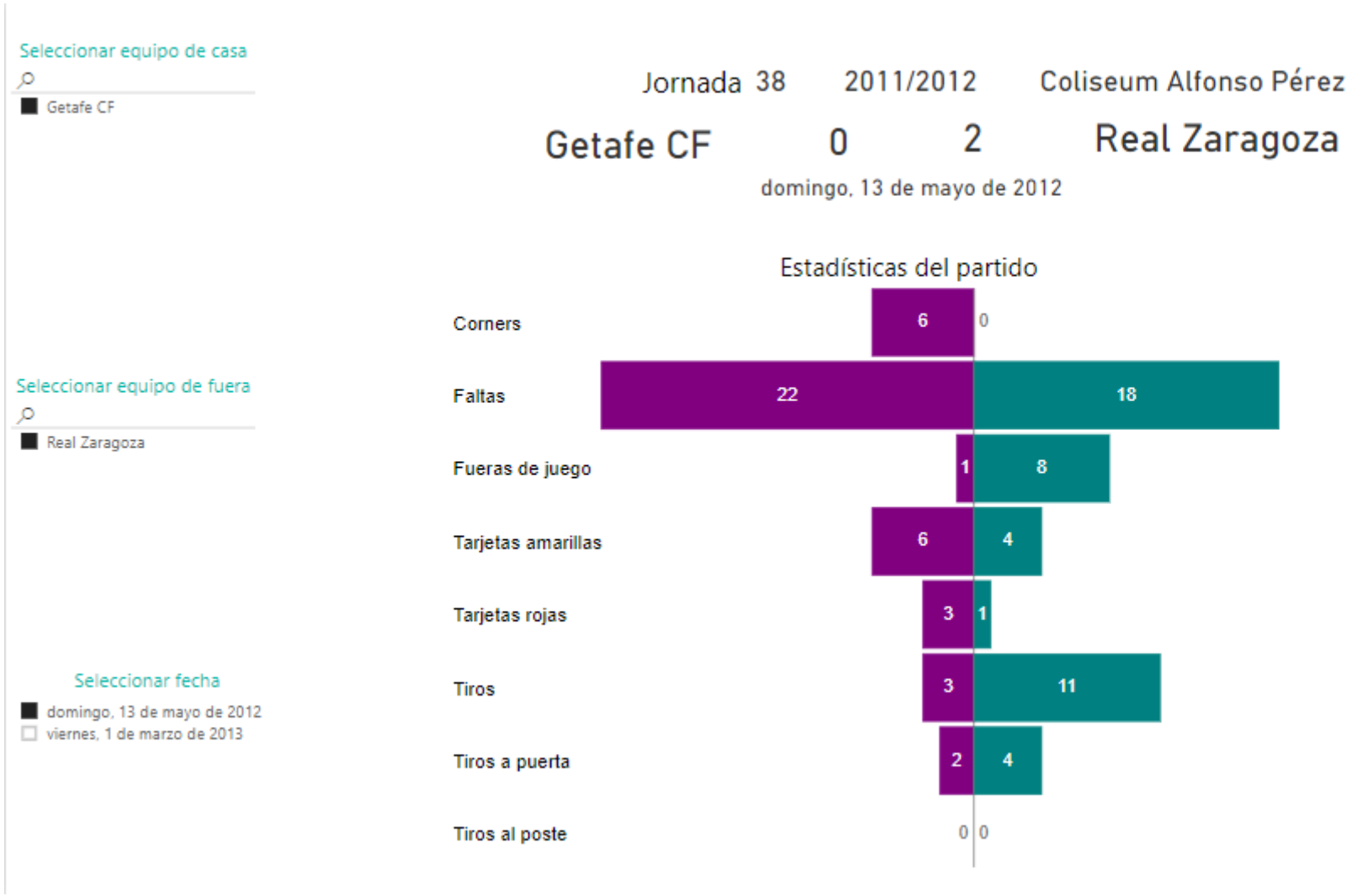

Figura E.2: Cuadro de estadísticas de los partidos.

E.3. Muestra de alineaciones

El cuadro de mandos "Muestra de alineaciones", de la Figura E.3, además de información básica del partido seleccionado por el usuario, presenta la alineación y los jugadores que han salido desde el banquillo en el mismo. Para seleccionar el partido, el usuario ha de elegir ambos equipos y la fecha (conforme se va seleccionando los demás se van filtrando, de forma que no es posible elegir algo que no existe y se aumenta la legibilidad).

La alineación se ha conseguido mostrar gracias al objeto visual del *marketplace* "EnhancedScatter" [74], que muestra valores en un gráfico en función de otros dos valores, los de los ejes. Al igual que con el cuadro de estadísticas, ha sido necesario darle un formato diferente a los datos, y para ello se ha utilizado KNIME. En este caso, el objeto requiere de una columna donde esté el nombre del jugador, otra con el valor del eje X y otro del eje Y, de forma que cada tupla contiene un jugador concreto en un partido concreto, sin distinción de si el jugador es del equipo local o del contrario.

Al igual que en el caso anterior, se ha empezado con una serie de operaciones *join* partiendo de AggFact_Partido para tener la información sobre los partidos necesaria. A continuación, se realiza una serie de operaciones *join* para tener almacenados en la tabla los nombres de todos los jugadores participantes en el partido y se realiza un *unpivoting*. El siguiente paso es tratar las coordenadas: se modifican las del eje X de los jugadores visitantes para que aparezcan en la otra mitad del campo en el gráfico, se incrementan en uno las coordenadas del eje X de los jugadores de casa para que se muestren mejor, y se invierten los jugadores (eje Y) del equipo visitante. El resultado se almacena en un fichero CSV llamado 'Info_Alineacion', utilizado finalmente en el gráfico. Para que se muestre el campo de fútbol debajo de los jugadores, en la pestaña de estilo de la gráfica se ha incluido como fondo una imagen encontrada en Internet [71].

Estos dos últimos cuadros de mandos se podrían haber integrado en uno solo. No obstante, se ha decidido separarlos por razones de espacio (quedarían muy pequeños si se incluyeran ambos en un solo cuadro) y por razones de rendimiento (cada partido tendría demasiadas tuplas, ya que para ambas gráficas se ha de desgranar distinta información en diferentes tuplas).

Figura E.3: Cuadro de muestra de las alineaciones.

E.4. Aptitudes de jugadores

El último cuadro de mandos, de la Figura E.4, "Aptitudes de jugadores", permite seleccionar un jugador y ver toda la información sobre sus aptitudes actuales. Esta va desde su información básica a su valoración general y potencial (sobre 100), pasando por sus aptitudes divididas en las de ataque, las de defensa y las de portería.

Para estas últimas se ha vuelto a utilizar un objeto visual extra del *marketplace*, "Radar Chart" [75]. Este objeto permite presentar de una forma visual valores de características de manera que dos sujetos sean fácilmente comparables. También requiere de parejas clave-valor, por lo que cada tupla representa un atributo de un jugador. Esta vez se ha decidido cambiar de método y utilizar PowerQuery, aprendiendo a manejarlo y empezando a conocer sus capacidades. Para conseguirlo, primero se ha duplicado la tabla de dimensión DIM_Jugador, llamándola Info_Jugador. A ésta se le han realizado dos operaciones *join*: con DIM_VersionJugador con el atributo versionActual y con DIM_DatosJugador, con el atributo idDatosJugador. A continuación, se ha realizado tres veces la llamada 'Anulación de dinamización de columnas', equivalente a un *unpivot*. Se ha realizado tres veces, ya que hay tres tipos de atributos: de portería, de defensa y de ataque. Los atributos de jugadores de campo se han dividido en estas dos últimas arbitrariamente. Así, y tras cambiar nombres de columnas, de valores y eliminar columnas innecesarias, ya se podían utilizar los radares. Sin embargo, como cada radar utilizaba como valor máximo el mayor valor entre los existentes de esos atributos en dicho jugador completo, los radares no seguían la misma escala para los diferentes jugadores, haciendo más complicado compararlos de forma visual. Para remediarlo, se ha creado una tabla llamada Info_Jugador_Aux a partir de la anterior, con una tupla por atributo donde el valor de todos ellos es 100 (valor máximo de los atributos). De esta forma, se consigue que todos los radares para todos los jugadores sigan la misma escala y se puedan comparar. Para que no sea visible en los radares, su contorno se ha puesto blanco.

Para realizar el gráfico de valoración general y potencial, se ha utilizado el objeto visual básico 'Medidor', cuyo valor es la valoración general del jugador, el valor máximo es 100 (se ha utilizado uno de los atributos que vale 100 en Info_Jugador_Aux) y el valor de destino es el potencial. De esta forma, es fácil ver la valoración actual del jugador y su potencial (el valor al que aspira).

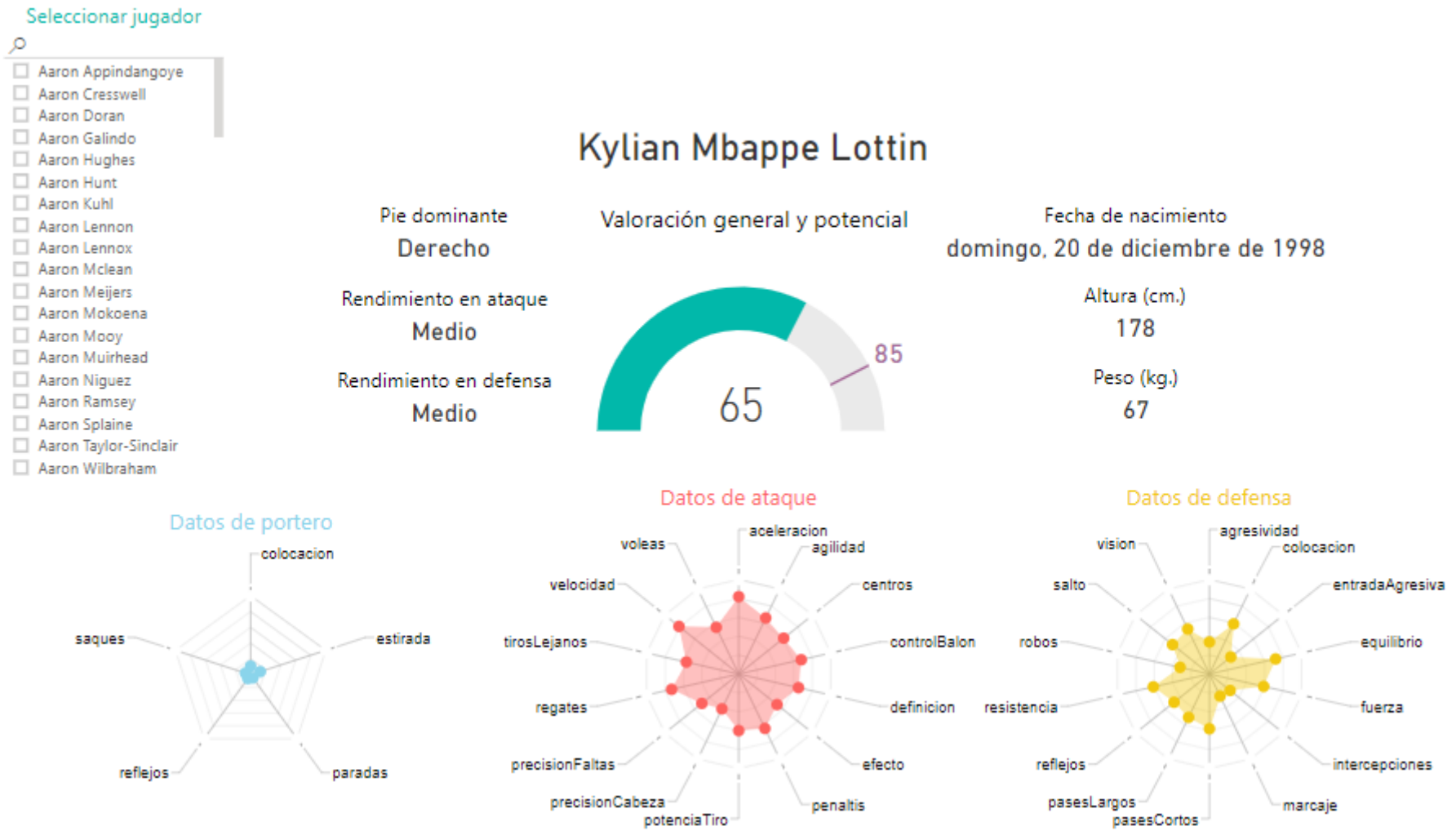

Figura E.4: Cuadro de aptitudes de los jugadores.

Anexo F

Minería de datos

En este anexo se explica y muestra todo lo relacionado con las técnicas de minería de datos aplicadas, ampliando la información de la Sección 5.3. En el Anexo F.1 se muestran ejemplos de funcionamiento y alineaciones de todos los métodos para un mismo partido. En el Anexo F.2 se explica lo relacionado a la evaluación de dichos métodos. En el Anexo F.3 se explica el proceso de selección de qué partidos utilizar para crear el modelo de regresión logística. Todo el código fuente del programa está accesible en el repositorio de GitHub del proyecto [53].

F.1. Traza del programa y ejemplos de alineaciones

En esta sección se va a mostrar tanto la traza de un ejemplo de funcionamiento por parte de un entrenador como un ejemplo de distintas alineaciones obtenidas según los distintos métodos y *baselines* planteados.

Primero, se va a mostrar una traza habitual del programa que podría utilizar el entrenador de un equipo para obtener una alineación para un partido concreto. En el programa, lo primero que ha de hacer el entrenador es buscar su equipo entre los que se le muestran ordenados alfabéticamente, introducir el id que aparece asociado al mismo y confirmar su elección. A continuación, ha de introducir la fecha del partido para el que desea la alineación. En caso de introducir una fecha en la que el equipo seleccionado no ha jugado ningún partido, ha de volver a introducir una fecha. Cuando introduce una correcta, obtiene *feedback* con información básica del partido. En ese momento, el entrenador puede elegir entre obtener la mejor alineación posible u obtener grupos de jugadores de su equipo que juegan muy bien juntos. En el caso de la alineación, se le mostraría la misma y se le darían las opciones finales. En el caso de los jugadores, se le pregunta si desea obtener defensas, delanteros o ambos. Cuando decide, los obtiene y el programa le da la opción de obtener una alineación a partir de dichos jugadores.

Posteriormente, se le muestran las opciones finales. Éstas permiten volver a obtener información del mismo partido, cambiar de partido, cambiar de equipo y partido, y finalizar. En el ejemplo mostrado en la Figura F.1, se ha obtenido la alineación que había de utilizar el Real Zaragoza contra el Real Betis el 26 de mayo de 2013.

A continuación, se mostrarán todos los equipos disponibles con sus respectivos ids, para que introduzca el de su equipo.
nombre idEquipo
1. FC Kaiserslautern 27
1. FC Köln 93
[...]
Wolverhampton Wanderers F.C. 71
Xerez C.D. 125
Introduzca el id de su equipo: **34**
Ha seleccionado Real Zaragoza (cuyo id es 34). ¿Es correcto? (Y/N): **Y**
Introduzca el día del partido que quiere analizar (dd): **26**
Introduzca el mes del partido que quiere analizar (mm): **05**
Introduzca el año del partido que quiere analizar (aaaa): **2013**
Ha elegido el partido Real Betis Balompié - Real Zaragoza, perteneciente a la jornada 37 de la temporada 2012/2013.
Si desea obtener la mejor alineación posible, escriba 'A'.
Si desea obtener los mejores grupos de jugadores de su equipo, escriba 'J': **J**
¿Desea recibir grupos de defensas ('DEF'), delanteros ('DEL') o ambos ('AMB')?: **DEF**
Los mejores defensas de su equipo son Cristian Sapunaru y Glenn Loovens.
¿Desea obtener una alineación completa a partir de estos jugadores? (Y/N): **Y**
Con esos jugadores, la mejor alineación sería: Roberto - Cristian Sapunaru - Glenn Loovens - Alvaro Gonzalez Soberon - Abraham Minero - Antonio Apono - Jose Movilla - Franco Zuculini - Victor Romero Rodriguez - Paco Montanes - Helder Postiga.
Si desea cambiar de equipo y de partido, escriba 'E'.
Si desea finalizar, escriba 'F'.
Si desea obtener información de otro partido, escriba 'P'.
Si desea obtener información del mismo partido pulse cualquier otra tecla: **F**

Figura F.1: Traza de ejemplo del programa de usuario.

Por otra parte, se ha realizado una prueba en la que se han obtenido alineaciones con todos los métodos y *baselines* creados, para comprobar las diferencias entre los mismos y con la alineación que se utilizó en realidad. Se ha utilizado "El Clásico" jugado en el Camp Nou en la temporada 2014/2015, obteniendo las alineaciones para ambos equipos. Se puede comprobar en las Figuras F.2 y F.3 que, excepto las alineaciones al azar, todas ellas tienen sentido (tanto a nivel de estructura como a nivel de calidad de los jugadores) si se conoce a sus protagonistas.

Su equipo es FC Barcelona. Ha elegido el partido FC Barcelona - Real Madrid C.F., perteneciente a la jornada 28 de la temporada 2014/2015.

El once real que se utilizó en ese partido fue:
Claudio Bravo - Daniel Alves - Gerard Pique - Jeremy Mathieu - Jordi Alba - Ivan Rakitic - Javier Mascherano - Andres Iniesta - Lionel Messi - Luis Suarez - Neymar.

El once obtenido por el *baseline* al azar tiene 7 diferencias, y es:
Claudio Bravo - Jeremy Mathieu - Pedro Rodriguez - Neymar - Daniel Alves - Martin Montoya - Rafinha - Munir El Haddadi - Sandro Ramirez - Sergio Busquets - Marc Bartra.

El once obtenido por el *baseline* de más usados tiene 2 diferencias, y es:
Claudio Bravo - Lionel Messi - Rafinha - Jordi Alba - Daniel Alves - Neymar - Sergio Busquets - Gerard Pique - Andres Iniesta - Ivan Rakitic - Luis Suarez.

El once obtenido por el *baseline* de más victorias tiene 2 diferencias, y es:
Claudio Bravo - Daniel Alves - Gerard Pique - Jeremy Mathieu - Jordi Alba - Ivan Rakitic - Sergio Busquets - Andres Iniesta - Pedro Rodriguez - Lionel Messi - Neymar.

El once obtenido por el método de mejor resultado, con datos de la misma temporada, tiene 1 diferencia, y es:
Claudio Bravo - Daniel Alves - Gerard Pique - Jeremy Mathieu - Jordi Alba - Ivan Rakitic - Sergio Busquets - Andres Iniesta - Lionel Messi - Luis Suarez - Neymar.

El once obtenido por el método de mejor resultado, con datos de todas las temporadas, tiene 3 diferencias, y es:
Claudio Bravo - Daniel Alves - Gerard Pique - Javier Mascherano - Jordi Alba - Xavi Hernandez - Sergio Busquets - Andres Iniesta - Pedro Rodriguez - Lionel Messi - Neymar.

El once obtenido por el método de regresión logística tiene 5 diferencias, y es:
Claudio Bravo - Lionel Messi - Jordi Alba - Marc Bartra - Rafinha - Xavi Hernandez - Neymar - Andres Iniesta - Ivan Rakitic - Adriano - Munir El Haddadi.

Figura F.2: Obtención de alineaciones del FC Barcelona para un partido.

F.2. Evaluación del rendimiento de los métodos de minería de datos

Para evaluar el rendimiento de los distintos métodos (así como de los *baselines*), se han realizado ocho ejecuciones, surgidas de las combinaciones de cuatro rangos de jornadas distintos y dos formas distintas de medir el número de diferencias con la

Su equipo es Real Madrid C.F.. Ha elegido el partido FC Barcelona - Real Madrid C.F., perteneciente a la jornada 28 de la temporada 2014/2015.

El once real que se utilizó en ese partido fue:
Iker Casillas - Daniel Carvajal - Pepe - Sergio Ramos - Marcelo - Luka Modric - Toni Kroos - Isco - Gareth Bale - Karim Benzema - Cristiano Ronaldo.

El once obtenido por el *baseline* al azar tiene 8 diferencias, y es:
Keylor Navas - James Rodriguez - Raphael Varane - Sergio Ramos - Jese Rodriguez - Gareth Bale - Sami Khedira - Alvaro Arbeloa - Lucas Silva - Cristiano Ronaldo - Martin Odegaard.

El once obtenido por el *baseline* de más usados tiene 1 diferencia, y es:
Iker Casillas - Toni Kroos - Gareth Bale - Cristiano Ronaldo - Karim Benzema - Marcelo - Daniel Carvajal - James Rodriguez - Isco - Pepe - Sergio Ramos.

El once obtenido por el *baseline* de más victorias tiene 1 diferencia, y es:
Iker Casillas - Daniel Carvajal - Pepe - Sergio Ramos - Marcelo - Luka Modric - Toni Kroos - James Rodriguez - Gareth Bale - Karim Benzema - Cristiano Ronaldo.

El once obtenido por el método de mejor resultado, con datos de la misma temporada, tiene 1 diferencia, y es:
Iker Casillas - Daniel Carvajal - Pepe - Sergio Ramos - Marcelo - Luka Modric - Toni Kroos - James Rodriguez - Gareth Bale - Karim Benzema - Cristiano Ronaldo.

El once obtenido por el método de mejor resultado, con datos de todas las temporadas, tiene 3 diferencias, y es:
Iker Casillas - Alvaro Arbeloa - Pepe - Sergio Ramos - Marcelo - Sami Khedira - Toni Kroos - James Rodriguez - Gareth Bale - Cristiano Ronaldo - Karim Benzema.

El once obtenido por el método de regresión logística tiene 4 diferencias, y es:
Pepe - Nacho Fernandez - Keylor Navas - Javier Hernandez - Karim Benzema - Gareth Bale - Lucas Silva - Cristiano Ronaldo - Isco - Sergio Ramos - Toni Kroos.

Figura F.3: Obtención de alineaciones del Real Madrid en un partido.

alineación real. En todas ellas se han utilizado un total de cien partidos aleatorios entre todos los posibles por rangos de jornadas. Nótese que cien partidos se refiere a cien equipos en cien partidos, ya que por cada partido se pueden obtener alineaciones de dos equipos diferentes.

Los cuatro rangos de jornadas son 1-13 (tercio inicial de la temporada), 14-25

(tercio central), 26-38 (tercio final) y 1-38 (toda la temporada). Con esta división, se pueden observar las diferencias existentes en los resultados, así como en el tiempo utilizado, mayor en el caso del tercio inicial. Esto se debe a que, dado que algunos equipos todavía no han ganado ningún partido, se han de utilizar varias técnicas hasta conseguir once jugadores. Para optimizar esto ligeramente, se han pasado como parámetro listas de jugadores ya obtenidos en la propia evaluación de los métodos anteriores para minimizar el número de repeticiones de la misma técnica en la misma evaluación.

En cuanto a las dos formas de medir las diferencias, tal y como se explicó en la Sección 5.3.2, éstas son la perfecta y la no perfecta. La primera es simple: la diferencia es el número de jugadores diferentes de la alineación obtenida respecto a la real. Su nombre surge de que se entiende que la alineación real es "perfecta", es decir, ganara o no el partido se entiende que el entrenador sacó la mejor alineación posible. Por otra parte, la no perfecta utiliza distintos ponderadores en función del resultado de dicho partido, de forma que un empate o una derrota hacen que la alineación real se considere menos relevante que con una victoria. Estos ponderadores son modificables porque se consideran diferentes para un club grande que para un club pequeño (un empate para el Real Madrid contra el Real Valladolid puede considerarse un fracaso, mientras que al contrario puede resultar un éxito). Por ello, estos ponderadores los puede cambiar fácilmente el administrador del programa. Por defecto son 1 para victoria, 0.6 para empate y 0.3 para derrota (no es 0 ya que hay variables externas más allá de la alineación que deriven en derrota, y equivaldría a ignorar el partido).

En las gráficas de las diferencias (Figuras F.4 a F.11) se puede comprobar que el *baseline* al azar es el peor que funciona (algo lógico, así como que funciona mejor cuanto más corta es la plantilla del equipo en la temporada, ya que tanto el entrenador como el algoritmo tienen menos jugadores entre los que elegir, por lo que las alineaciones se asemejarán más) y los que mejor funcionan son los métodos de mejores resultados, tanto con datos de la misma temporada como con los de todas. El método de regresión logística es bastante variable y funciona peor que estos dos, algo que se puede explicar por los pocos datos con los que trabaja, por lo que el modelo no se ajusta tanto como se desearía. También cabe destacar que los otros dos *baselines* (los que utilizan a los más usados y los que han participado en más victorias individualmente) funcionan bastante bien. La explicación a esto es que los entrenadores suelen ser bastante "conservadores", de forma que varían muy poco sus alineaciones, independientemente de sus resultados.

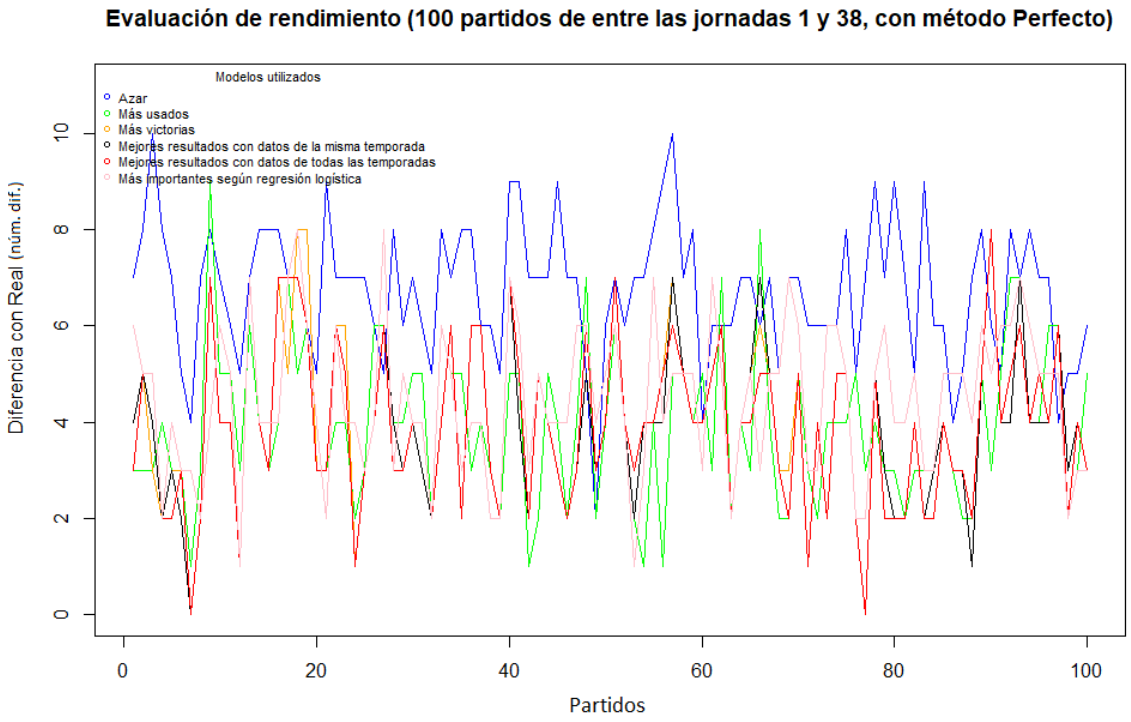

Figura F.4: Evaluación de rendimiento con 100 partidos entre las jornadas 1 y 38 con método perfecto.

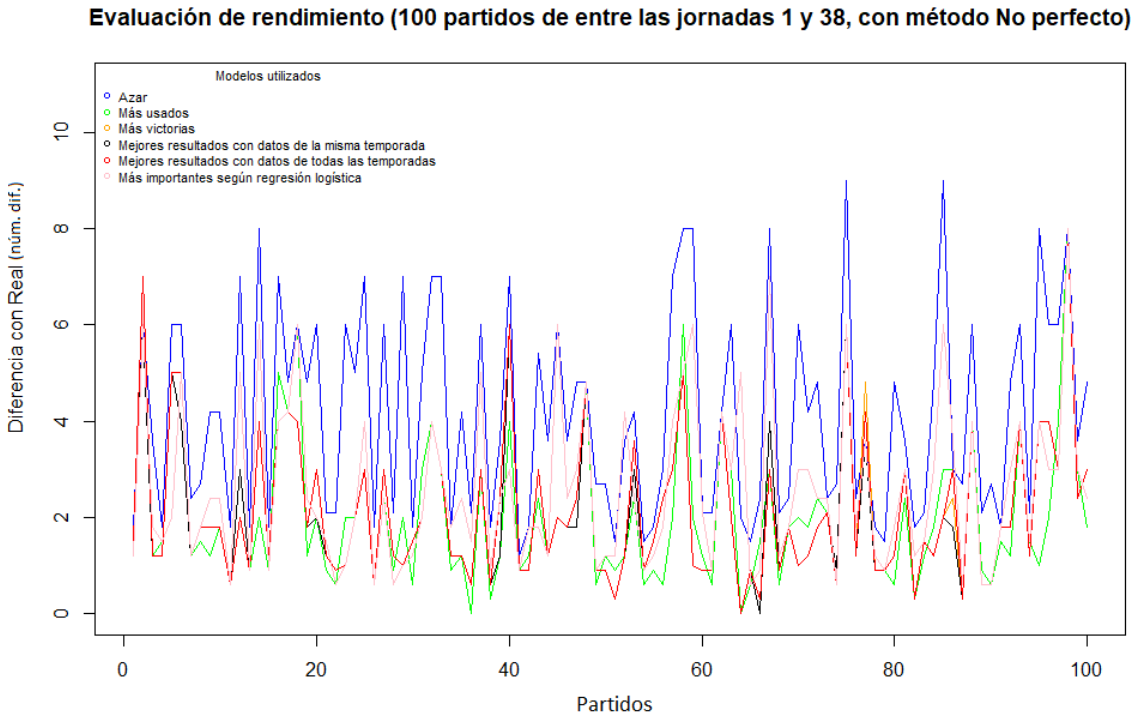

Figura F.5: Evaluación de rendimiento con 100 partidos entre las jornadas 1 y 38 con método no perfecto.

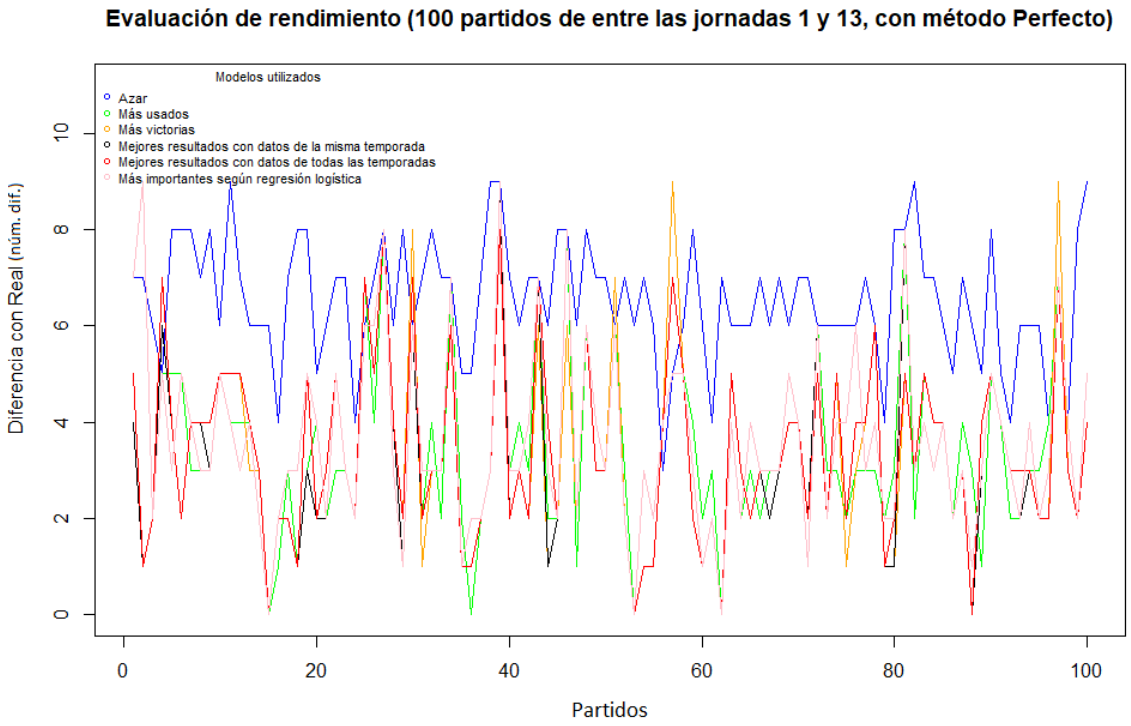

Figura F.6: Evaluación de rendimiento con 100 partidos entre las jornadas 1 y 13 con método perfecto.

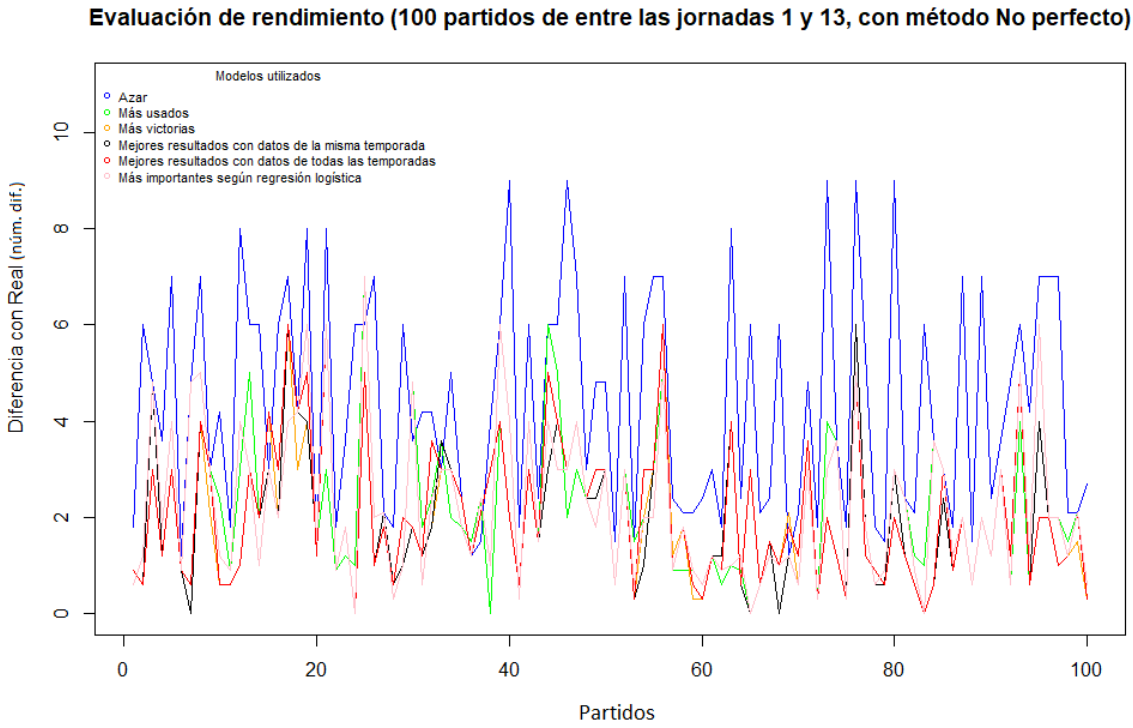

Figura F.7: Evaluación de rendimiento con 100 partidos entre las jornadas 1 y 13 con método no perfecto.

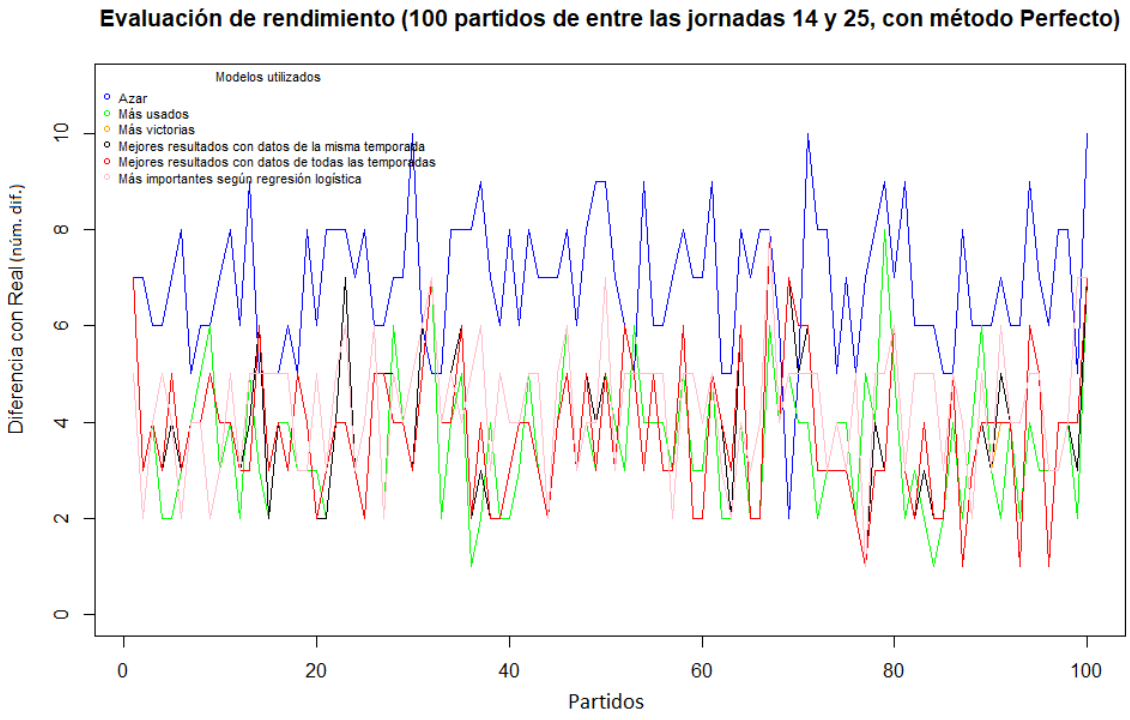

Figura F.8: Evaluación de rendimiento con 100 partidos entre las jornadas 14 y 25 con método perfecto.

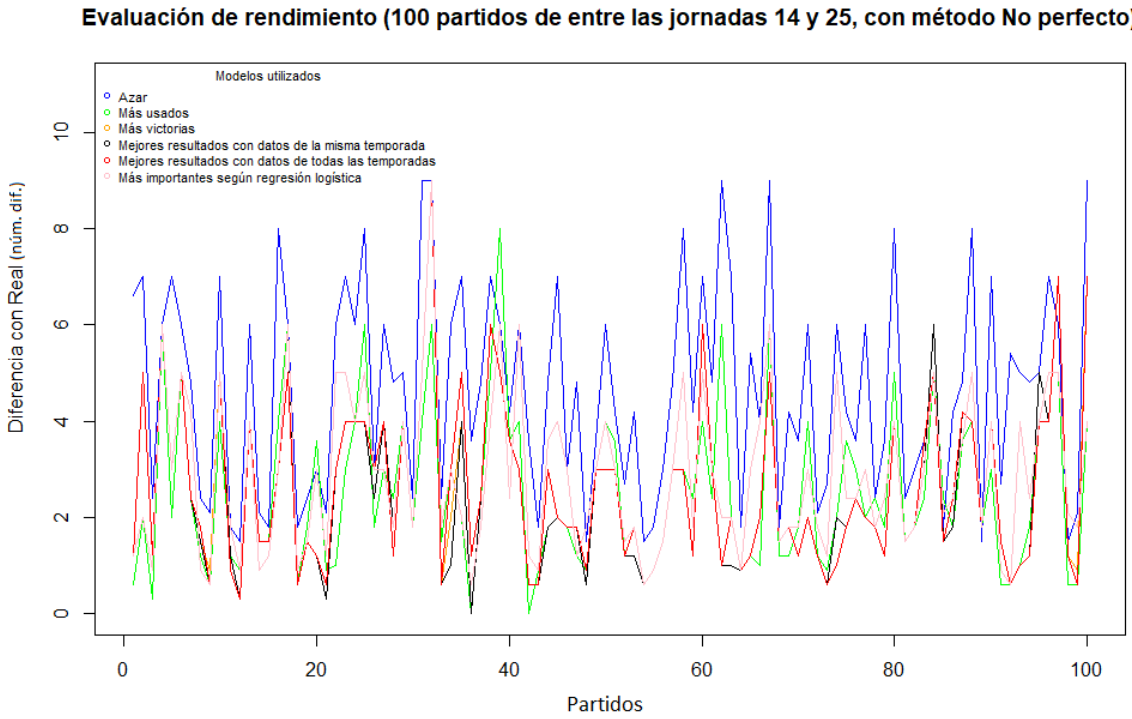

Figura F.9: Evaluación de rendimiento con 100 partidos entre las jornadas 14 y 25 con método no perfecto.

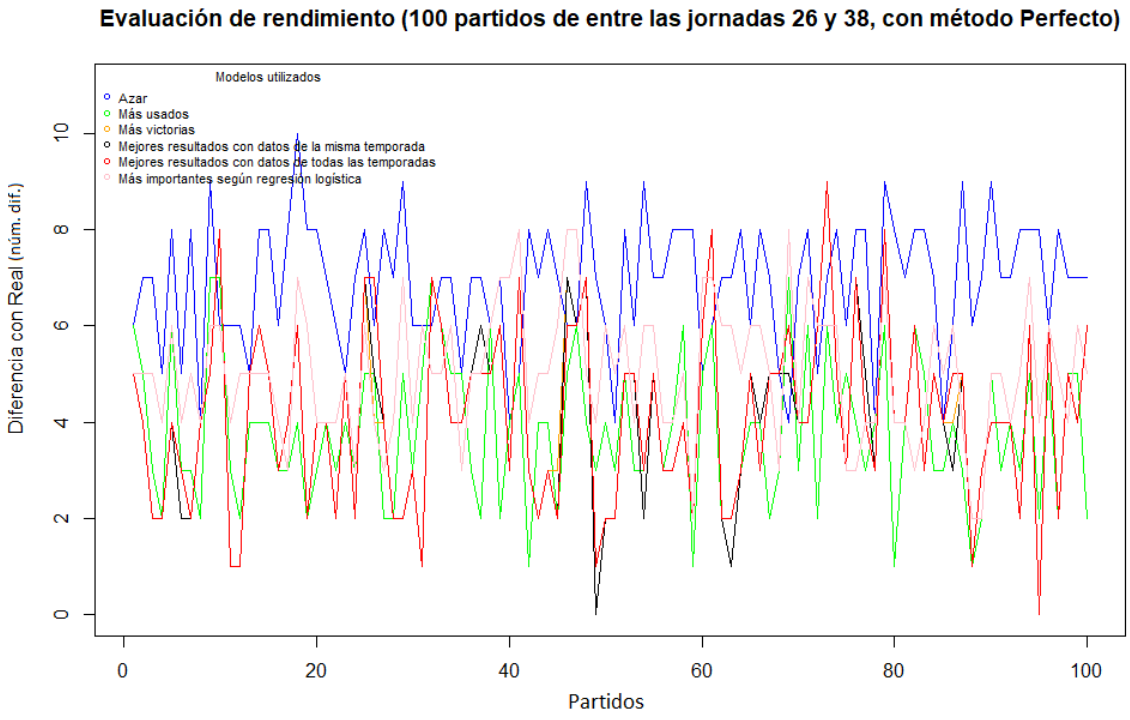

Figura F.10: Evaluación de rendimiento con 100 partidos entre las jornadas 26 y 38 con método perfecto.

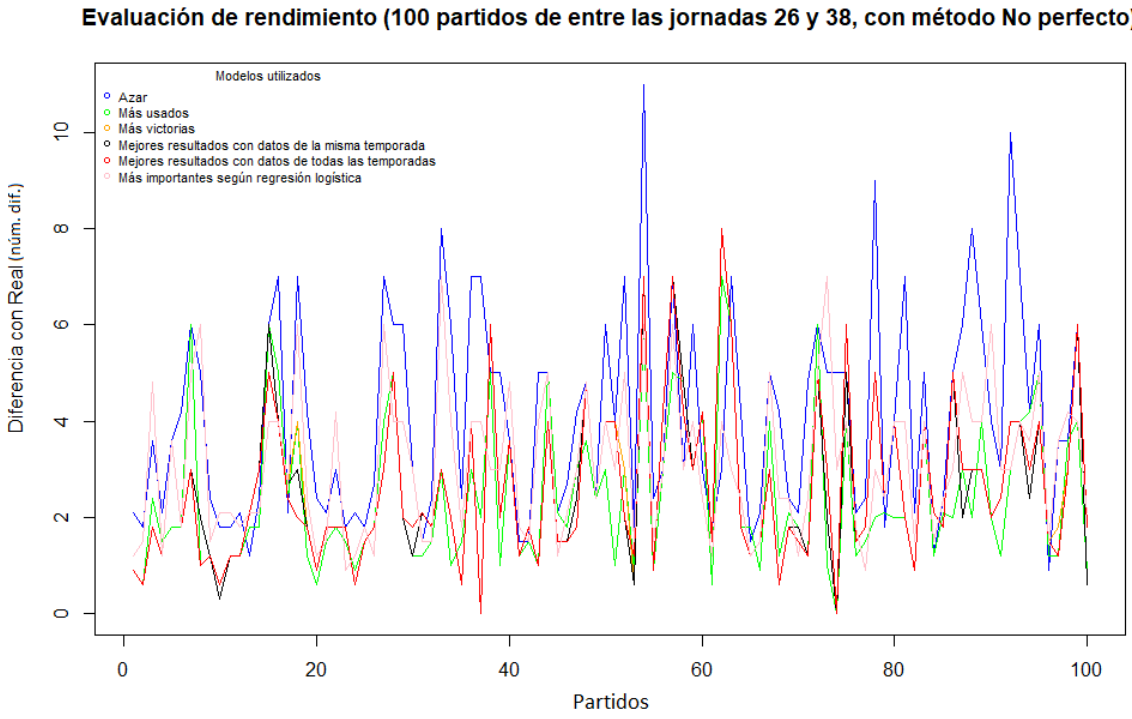

Figura F.11: Evaluación de rendimiento con 100 partidos entre las jornadas 26 y 38 con método no perfecto.

En cuanto a los tiempos empleados por los distintos métodos, los únicos significativos son los de mejores resultados (tanto con la misma temporada como con todas), especialmente en las primeras jornadas, dado que en ocasiones necesita repetir las técnicas varias veces con distintos parámetros para conseguir obtener un once, ya que los equipos involucrados no habían conseguido ni una victoria ni probablemente un empate todavía. No hay correlación entre el tiempo empleado y el número de datos usados. En las Tablas F.1, F.2 y F.3 se pueden observar los valores de los errores absolutos medios, errores cuadráticos medios y tiempos medios para las anteriores ejecuciones, respectivamente. Las gráficas resumidas de estos valores se pueden observar en la Sección 5.3 y los acrónimos empleados se pueden consultar en la tabla 5.1.

Jor.	Perfecto						No Perfecto					
	AZ	MU	MV	MRM	MRT	RL	AZ	MU	MV	MRM	MRT	RL
1-38	6.7	3.9	4.0	3.9	3.9	4.5	4.1	2.0	2.1	2.1	2.2	2.6
1-13	6.5	3.5	3.3	3.3	3.4	3.6	4.4	2.2	1.9	2.0	2.0	2.2
14-25	6.9	3.6	3.8	3.8	3.8	4.3	4.6	2.5	2.5	2.5	2.6	2.9
26-38	6.9	3.8	4.1	4.1	4.1	5.0	4.1	2.4	2.6	2.6	2.6	3.1

Tabla F.1: Errores absolutos medios de los experimentos.

Jor.	Perfecto						No Perfecto					
	AZ	MU	MV	MRM	MRT	RL	AZ	MU	MV	MRM	MRT	RL
1-38	6.9	4.2	4.3	4.3	4.3	4.7	4.7	2.5	2.6	2.6	2.7	3.1
1-13	6.7	4.0	3.9	3.9	3.9	4.1	4.9	2.7	2.4	2.5	2.5	2.8
14-25	7.0	3.9	4.1	4.1	4.1	4.5	5.1	3.0	3.1	3.1	3.1	3.3
26-38	7.0	4.1	4.5	4.5	4.5	5.2	4.6	2.9	3.1	3.1	3.1	3.5

Tabla F.2: Errores cuadráticos medios de los experimentos.

Jor.	Perfecto						No Perfecto					
	AZ	MU	MV	MRM	MRT	RL	AZ	MU	MV	MRM	MRT	RL
1-38	0.2	0.5	0.4	16.4	8.3	0.2	0.2	0.5	0.3	29.0	3.3	0.2
1-13	0.3	0.7	0.5	121.4	16.6	0.2	0.2	0.6	0.3	177.0	36.1	0.12
14-25	0.2	0.6	0.4	2.4	1.6	0.2	0.2	0.5	0.3	2.0	1.0	0.2
26-38	0.2	0.5	0.4	0.9	0.9	0.3	0.2	0.5	0.3	1.2	0.9	0.3

Tabla F.3: Tiempos medios de los experimentos (en segundos).

F.3. Elección de partidos para regresión logística

Para crear el modelo de la regresión logística, a la hora de qué partidos utilizar, se podía plantear de tres formas distintas: solo con las victorias del equipo en la presente temporada, con las victorias y los empates, y con todos sus partidos (victorias, empates y derrotas). Para ponderar la importancia del resultado, las victorias siempre valdrían 1, los empates 0.5 y las derrotas 0. Para decidir cuál de las tres formas se utilizaría finalmente, se compararon los modelos obtenidos con todas ellas en un partido concreto (FC Barcelona - Deportivo de La Coruña, del 23 de mayo de 2015, perteneciente a la última jornada de liga, ya que es cuando de más información se dispone). En las Figuras F.12, F.13 y F.14 se muestran los parámetros de validación de los distintos modelos, de forma que comparándolos entre sí se puede comprobar qué modelo funciona mejor.

```
    Null deviance: 0.0000e+00  on 26  degrees of freedom
Residual deviance: 4.2579e-10  on  6  degrees of freedom
AIC: 42

Number of Fisher Scoring iterations: 24
```

Figura F.12: Parámetros de validación con victorias.

```
    Null deviance: 1.2838e+01  on 33  degrees of freedom
Residual deviance: 3.6840e-10  on 12  degrees of freedom
AIC: 53.704

Number of Fisher Scoring iterations: 25
```

Figura F.13: Parámetros de validación con victorias y empates.

```
    Null deviance: 24.689  on 36  degrees of freedom
Residual deviance:  4.201  on 15  degrees of freedom
AIC: 58.724

Number of Fisher Scoring iterations: 22
```

Figura F.14: Parámetros de validación con victorias, empates y derrotas.

Se puede comprobar que en todos ellos el *deviance* disminuye sustancialmente (lo que indica que la medida del error cometido es mucho menor con los predictores que sin ellos), aunque especialmente en el que solo se utilizan las victorias (Figura F.12), que a su vez es el modelo con un AIC (*Akaike Information Criterion* [78]) más bajo, indicando que es el modelo que mejor se ajusta. Por estas razones se han utilizado únicamente las victorias de forma predeterminada, aunque la función creada en R permite utilizar cualquiera de las tres formas.